

三菱电机安全可编程控制器

MELSEC **QS** series

QSCPU用户手册 (功能解说/程序基础篇)

-QS001CPU



● 安全注意事项 ●

(在使用之前请务必阅读本说明

使用本产品之前，应仔细阅读本手册以及本手册介绍的相关手册，同时在充分注意安全的前提下，进行正确的操作。

本手册中，安全注意事项被分为“危险”和“注意”这二个等级。



危险

表示错误操作可能造成灾难性后果，引起死亡或重伤事故。



警告

表示错误操作可能造成危险后果，引起人员中等伤害或轻伤还可能使设备损坏。

注意根据情况不同，即使  这一级别的事项也有可能引发严重后果。

对两级注意事项都须遵照执行，因为它们对于操作人员安全是至关重要的。

妥善保管本手册，放置于操作人员易于取阅的地方，并应将本手册交给最终用户。

[设计方面的注意事项]



危险

- 安全可编程控制器在检测出外部电源异常或可编程控制器主机故障时将使输出为 OFF。应设置外部电路，确保通过可编程控制器的输出 OFF 可以可靠地停止危险源的动力。
如果电路设置不正确，有可能会引发事故。
- 应在安全可编程控制器的外部设置安全继电器的短路电流保护、保险丝、断路器等保护电路。
- 应在顺控程序及安全可编程控制器的外部设置连锁电路，以确保通过个人计算机对运行中的安全可编程控制器进行数据变更、程序变更及状态控制时，整个系统能够安全运行。
在操作安全可编程控制器时，应熟读手册，预先确定操作步骤，充分确认安全后再进行操作。
此外，在通过个人计算机对安全 CPU 模块进行在线操作时，应预先制定由于电缆接触不良等导致发生通信异常时的系统处理方法。
- 从安全 CPU 模块至 CC-Link Safety 系统主站模块的输出信号 (Y) 全部为“禁止使用”。
关于“禁止使用”信号的有关内容请参阅 CC-Link Safety 系统主站模块用户手册 (详细篇)。
如果对这些信号进行 ON/OFF，有导致可编程控制器系统误动作的危险。
此外，由于无法保证正常动作，因此不要通过顺控程序进行 ON/OFF。

[设计方面的注意事项]

危险

- 检测出 CC-Link Safety 异常的安全远程 I/O 模块的输出将为 OFF。
顺控程序的输出不能自动 OFF。
应编制当检测出 CC-Link Safety 异常时使输出 OFF 的顺控程序。
如果在输出 ON 的状态下复原 CC-Link Safety，由于设备的突然动作，有可能会引发事故。
- 应编制一个连锁程序，确保当安全功能动作使输出为 OFF 后，只有通过复位按钮等手动操作才可以重新启动。

注意

- 外部设备的连线和通信电缆请勿与主电路及动力电缆捆扎在一起，也勿使其相距过近。应大约隔开 100mm 以上。因为噪声会引起误动作。

[安装方面的注意事项]

注意

- 安全可编程控制器应在本手册中规定的一般技术规格的环境下使用。
如果在一般技术规格范围以外的环境下使用，会引起触电、火灾、误动作、产品损坏或者性能劣化现象的发生。
- 安装模块时应在按住模块下部的安装卡子的同时，将模块固定用突起物可靠地插入基板的固定孔，并以固定孔为支点进行安装。
如果模块没有得到的正确安装，则会引起误动作、故障及脱落。
必须应使用螺栓将模块牢固地安装在基板上。
紧固螺栓应在规定的扭矩范围内进行。
螺栓如果过松会引起脱落、短路、误动作。
螺栓如果过紧，会导致螺栓和模块的损坏从而引起脱落、短路以及误动作。
- 模块的拆装必须要在将系统中使用的外部电源全部切断之后进行。
如果不全部切断，就有损伤产品的危险。
- 不要直接接触模块的带电部分。
否则会引起模块误动作及发生故障。

[连线时的注意事项]

危险

- 连线作业等必须要在将系统中使用的外部电源全部切断之后进行。
不全部切断电源会有触电或者损伤产品的危险。
- 连线作业之后进行通电、运行时，必须在产品上安装附属的端子盖。
如果端子盖没有盖上的话，有触电的危险。

注意

- FG 端子以及 LG 端子必须可靠接地，其接地等级为可编程控制器专用的 D 种接地（第三种接地）以上。
否则会有触电、误动作的危险。
- 端子排连线应使用带绝缘套的压装端子。
此外，在一个端子上最多只能连接 2 个压装端子。
- 应使用合适的压装端子，并按规定的扭矩拧紧。
如果使用 Y 型端子，若端子螺栓松动将可能导致脱落、故障。
- 模块的连线必须在确认产品的额定电压以及端子排列之后正确地进行操作。
与额定电压相异的电源连接或者连线错误会导致火灾以及故障的发生。
- 端子排安装螺栓、端子螺栓及模块安装螺栓的紧固应在规定扭矩范围内进行。
端子螺栓如果过松则会引起短路、火灾以及误动作。
端子螺栓如果过紧，则可能由于螺栓和模块的损坏而引起脱落、短路以及误动作。
如果模块安装螺栓过松可能会导致脱落。
如果模块安装螺栓过紧，则可能由于螺栓和模块的损坏而引起脱落。
- 应注意模块内不要弄进切屑和连线碎块等异物。
否则会引起火灾、故障、误动作。
- 为了防止在连线时布线配件、碎块等异物进入模块内，在模块上部贴着防止杂物混入的贴纸。
在连线作业中不要揭下此贴纸。
在系统运行时，为了更好地散热，务必揭下此贴纸。
- 三菱公司的可编程控制器应安装在控制盘内使用。安装在控制盘内的可编程控制器的电源模块应通过中继端子排与主电源连线。
此外，在对电源模块进行更换及连线作业时，应由在触电保护方面受到过良好培训的维护人员进行操作。
关于连线方法请参阅 QSCPU 用户手册（硬件设计 / 维护点检篇）。

[启动、维护时的注意事项]



- 在通电状态下不要接触端子。
否则会有触电的危险。
- 应正确地连接电池。
不要对电池进行充电、分解、加热、扔进火中、短路以及焊接等操作。
如果对电池处理不当，由于电池发热、破裂、起火的原因，会引发火灾以及造成人员损伤。
- 在进行清扫或对端子排螺栓、端子螺栓、模块固定螺栓进行紧固作业之前必须先将系统中使用的外部电源全部切断。
不全部切断电源会有触电的危险。
端子排螺栓、端子螺栓、模块固定螺栓应在规定的扭矩范围内拧紧。
如果端子排螺栓、端子螺栓过松则会引起短路火灾以及误动作。
如果端子排螺栓、端子螺栓过紧，则可能由于螺栓和模块的破损而引起脱落、短路以及误动作。
如果模块固定螺栓过松会导致脱落。
如果模块固定螺栓过紧则可能由于螺栓和模块的破损而引起脱落。

[启动、维护时的注意事项]

注意

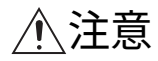
- 通过个人计算机对运行中的安全可编程控制器进行在线操作（安全 CPU 运行中的程序变更、软元件测试、RUN-STOP 等运行状态的变更）时，应在熟读手册、充分确认安全的基础上进行。
应由受过培训的操作人员按照设计时确定的操作步骤进行操作。
此外，在对安全 CPU 进行运行中的程序变更（运行中写入）时，根据操作条件有时会发生程序损坏的现象。
应在充分理解了 GX Developer 手册中记载的注意事项的基础上进行操作。
- 不要对各模块进行分解和改造。
否则会引起故障、误动作、人员受伤以及火灾的发生。
如果在除三菱公司或者三菱公司指定的 FA 中心以外的地方进行了修理及改造等，将不再作为质保对象。
- 手机等无线通信设备应在距安全可编程控制器主机周边 25cm 以上的距离使用。
否则可能导致误动作。
- 模块的拆装必须要在将系统中使用的外部电源全部切断之后进行。
如果不全部切断，就有损伤产品的危险。
模块、基板及端子排在投入使用后，其拆装次数应不超过 50 次。（根据 IEC61131-2 标准）
如果其拆装次数超过了 50 次，有可能导致误动作。
- 应防止安装在模块上的电池掉落以及受到撞击。
掉落以及受到撞击会使电池发生破损以及电池内部发生电池漏液。
掉落、受到撞击的电池不要使用并应将其废弃。
- 在接触模块之前，必须要先触摸已接地的金属，以放掉人体上所带的静电。
如果不放掉静电则会引起模块发生故障以及产生误动作。

[废弃时的注意事项]

注意

- 产品废弃的时候，应作为工业废品来处理。

[运输时的注意事项]



- 运输含有锂的电池时，必须按照运输规定进行处理。
(规定对象种类的详细内容请参阅附录 5。)

修订记录

* 手册编号在封底的左下角。

印刷日期	手册编号	修订记录
2007 年 08 月	SH(NA)-080713CHN-A	初版

日文手冊原稿：SH-080608-B

本手册不授予任何工业产权或任何其它类型的产权，也不授予任何专利许可。三菱电机对由于使用了本手册中的内容而引起的涉及工业知识产权的任何问题不承担责任。

序言

此次，非常感谢贵方购买了三菱安全可编程控制器 MELSEC-QS 系列。
在使用前请熟读本手册，并在充分理解 QS 系列可编程控制器的功能及性能的基础上正确地使用。

目录

第 1 章 概要	1 - 1 到 1 - 10
1.1 特点	1 - 2
1.2 程序的存储及运算	1 - 6
1.3 便于编程的软元件、指令	1 - 8
1.4 序列号及功能版本的确认方法	1 - 9
第 2 章 性能规格	2 - 1 到 2 - 2
第 3 章 顺控程序的执行	3 - 1 到 3 - 20
3.1 顺控程序	3 - 2
3.1.1 顺控程序的记述方法	3 - 3
3.1.2 顺控程序的运算	3 - 3
3.2 扫描时间的思路	3 - 5
3.3 运算处理	3 - 7
3.3.1 初始化处理	3 - 7
3.3.2 CC-Link Safety、MELSECNET/H 的刷新	3 - 7
3.3.3 I/O 刷新	3 - 8
3.3.4 END 处理	3 - 8
3.4 RUN 状态、STOP 状态的运算处理	3 - 9
3.5 瞬间掉电时的运算处理	3 - 11
3.6 数据的清除处理	3 - 12
3.7 顺控程序中可使用的数值	3 - 13
3.7.1 BIN(2 进制数 :Binary Code)	3 - 16
3.7.2 HEX(16 进制数 :Hexadecimal)	3 - 18
3.7.3 BCD(2 进制编码的 10 进制数 :Binary Coded Decimal)	3 - 19
第 4 章 I/O 地址号的分配	4 - 1 到 4 - 12
4.1 关于 I/O 地址号	4 - 1
4.2 I/O 地址号的分配思路	4 - 2
4.2.1 基板的 I/O 地址号	4 - 2
4.2.2 远程站的 I/O 地址号	4 - 3
4.3 通过 GX Developer 进行 I/O 分配	4 - 5
4.3.1 通过 GX Developer 进行 I/O 分配的目的	4 - 5
4.3.2 通过 GX Developer 进行 I/O 分配的思路	4 - 6

4.3.3 I/O 分配示例	4 - 10
4.4 I/O 地址号的确认	4 - 11

第 5 章 关于 CPU 模块中使用的存储器及文件	5 - 1 到 5 - 17
----------------------------------	-----------------------

5.1 CPU 模块中使用的存储器	5 - 1
5.1.1 存储器构成及可存储数据	5 - 1
5.1.2 关于程序存储器	5 - 3
5.1.3 关于内置 ROM	5 - 6
5.1.4 内置 ROM 的程序的执行（引导运行）及写入	5 - 8
5.2 程序文件的构成	5 - 13
5.3 通过 GX Developer 进行文件操作及处理时的注意事项	5 - 15
5.3.1 文件操作	5 - 15
5.3.2 文件处理时的注意事项	5 - 15
5.3.3 文件的容量	5 - 16
5.3.4 文件的大小单位	5 - 17

第 6 章 功能	6 - 1 到 6 - 61
-----------------	-----------------------

6.1 功能一览	6 - 1
6.2 安全 CPU 动作模式	6 - 2
6.2.1 安全 CPU 动作模式	6 - 2
6.2.2 安全 CPU 动作模式的确认	6 - 5
6.2.3 安全 CPU 动作模式的切换	6 - 7
6.2.4 安全 CPU 动作模式、CPU 动作状态的各功能的动作	6 - 13
6.2.5 可以通过 GX Developer 对 CPU 模块进行的在线操作	6 - 15
6.3 CPU 存取密码	6 - 17
6.4 可编程控制器存储器初始化	6 - 20
6.5 测试模式下的连续 RUN 的防止设置	6 - 24
6.6 至 ROM 的写入次数的确认	6 - 26
6.7 自诊断功能	6 - 27
6.7.1 出错时的 LED 显示	6 - 30
6.7.2 出错的解除	6 - 30
6.8 操作内容、自诊断出错发生内容的记录（操作 / 故障历史记录功能）	6 - 32
6.9 恒定扫描	6 - 36
6.10 STOP 状态 $\longleftrightarrow$ RUN 状态时的输出（Y）状态的设置	6 - 39
6.11 时钟功能	6 - 42
6.12 远程操作	6 - 45
6.12.1 远程 RUN/STOP	6 - 45
6.12.2 远程 RESET（远程复位）	6 - 48
6.12.3 远程操作与 CPU 模块的 RUN/STOP 状态的关系	6 - 50
6.13 监视功能	6 - 51
6.14 CPU 模块运行中的程序写入	6 - 52
6.14.1 梯形图模式下的运行中写入	6 - 52
6.15 看门狗定时器（WDT）	6 - 56

6.16 通过 GX Developer 进行 CPU 模块的系统显示	6 - 58
6.17 LED 的显示	6 - 60
6.17.1 LED 的熄灯方法	6 - 60
第 7 章 与智能功能模块的通信	7 - 1 到 7 - 2
7.1 与 CC-Link Safety 主站模块的通信	7 - 1
7.2 与 MELSECNET/H 模块的通信	7 - 2
第 8 章 参数	8 - 1 到 8 - 12
8.1 可编程控制器参数	8 - 2
8.2 网络参数	8 - 10
第 9 章 软元件说明	9 - 1 到 9 - 34
9.1 软元件一览	9 - 1
9.2 内部用户软元件	9 - 2
9.2.1 输入 (X)	9 - 5
9.2.2 输出 (Y)	9 - 7
9.2.3 内部继电器 (M)	9 - 8
9.2.4 报警器 (F)	9 - 9
9.2.5 变址继电器 (V)	9 - 13
9.2.6 链接继电器 (B)	9 - 14
9.2.7 链接特殊继电器 (SB)	9 - 16
9.2.8 定时器 (T)	9 - 17
9.2.9 计数器 (C)	9 - 23
9.2.10 数据寄存器 (D)	9 - 27
9.2.11 链接寄存器 (W)	9 - 28
9.2.12 链接特殊寄存器 (SW)	9 - 30
9.3 内部系统软元件	9 - 31
9.3.1 特殊继电器 (SM)	9 - 31
9.3.2 特殊寄存器 (SD)	9 - 32
9.4 嵌套 (N)	9 - 33
9.5 常数	9 - 34
9.5.1 10 进制数常数 (K)	9 - 34
9.5.2 16 进制数常数 (H)	9 - 34
第 10 章 CPU 模块的处理时间	10 - 1 到 10 - 8
10.1 扫描时间	10 - 1
10.1.1 扫描时间的构成及计算公式	10 - 1
10.1.2 扫描时间相关要素的处理时间	10 - 3
10.1.3 扫描时间的延长原因	10 - 5
10.2 其它处理时间	10 - 7
第 11 章 将程序写入 CPU 模块中的步骤	11 - 1 到 11 - 4

11.1	创建程序时的确认事项.....	11 - 1
11.2	程序的写入步骤.....	11 - 2
11.3	引导运行的步骤.....	11 - 4

附录	附录 - 1 到附录 - 24
----	-----------------

附录 1	特殊继电器一览.....	附录 - 1
附录 2	特殊寄存器一览.....	附录 - 5
附录 3	参数号一览.....	附录 - 19
附录 4	在安全 CPU 模块中使用 MELSECNET/H 模块时的限制	附录 - 21
附录 5	电池运输时的注意事项.....	附录 - 24

索引	索引 - 1 到索引 - 2
----	----------------

(相关手册) QSCPU 用户手册 (硬件设计 / 维护点检篇)

目录

第 1 章 概要

1.1 特点

第 2 章 系统配置

2.1 基本系统配置

2.2 外部设备的构成

2.3 序列号及功能版本的确认方法

第 3 章 一般规格

第 4 章 CPU 模块

4.1 性能规格

4.2 各部位的名称

4.3 程序写入时的开关操作

4.4 复位操作

第 5 章 电源模块

5.1 规格

5.2 不间断电源装置及连接时的注意事项

5.3 各部位的名称及设置

第 6 章 基板

6.1 规格一览

6.2 各部位的名称

第 7 章 电池

7.1 规格 (Q6BAT)

7.1.1 电池的安装

第 8 章 CPU 模块的启动步骤

8.1 安全模式下投运前的步骤

第 9 章 EMC 指令 / 低电压指令

9.1 符合 CEM 指令的要求

- 9.1.1 EMC 指令相关规格
- 9.1.2 控制盘内的安装
- 9.1.3 电缆
- 9.1.4 电源模块
- 9.1.5 其它

9.2 符合低电压指令的要求

- 9.2.1 MELSEC_QS 系列可编程控制器适用规格
- 9.2.2 MELSEC_QS 系列可编程控制器的选定
- 9.2.3 供应电源
- 9.2.4 控制盘
- 9.2.5 接地
- 9.2.6 外部配线

第 10 章 安装及设置

10.1 可编程控制器的发热量的计算方法

10.2 模块的安装

- 10.2.1 使用时的注意事项
- 10.2.2 基板的安装注意事项
- 10.2.3 模块的安装、拆卸

10.3 配线

- 10.3.1 配线注意事项
- 10.3.2 电源模块的配线

第 11 章 维护点检

11.1 日常点检

11.2 定期点检

11.3 电池的寿命及更换步骤

- 11.3.1 CPU 模块的电池寿命
- 11.3.2 CPU 模块的电池更换步骤

11.4 卸下电池保管后再次使用时

11.5 保管的电池超出寿命后使用时

第 12 章 故障排除

12.1 故障排除的基本

12.2 故障排除

- 12.2.1 故障排除的划分流程
- 12.2.2 ERR 端子 (负逻辑) 为 OFF (开放) 时的流程
- 12.2.3 “ POWER ” LED 熄灯时的流程
- 12.2.4 “ ALIVE ” LED 不亮灯 / 熄灯时的流程
- 12.2.5 “ RUN ” LED 熄灯时的流程
- 12.2.6 “ RUN ” LED 闪烁时
- 12.2.7 “ ERR. ” LED 亮灯 / 闪烁时的流程
- 12.2.8 “ USER ” LED 亮灯时
- 12.2.9 “ BAT. ” LED 亮灯时
- 12.2.10 不能读取程序时的流程
- 12.2.11 不能写入程序时的流程

12.3 出错代码一览

- 12.3.1 全部出错代码
- 12.3.2 出错代码的读取方法
- 12.3.3 出错代码一览 (1000 ~ 1999)
- 12.3.4 出错代码一览 (2000 ~ 2999)
- 12.3.5 出错代码一览 (3000 ~ 3999)
- 12.3.6 出错代码一览 (4000 ~ 4999)
- 12.3.7 出错代码一览 (5000 ~ 5999)
- 12.3.8 出错代码一览 (8000 ~ 9000)

12.4 出错解除

12.5 与 CPU 模块通信时返回至请求源的出错代码

12.6 特殊继电器一览

12.7 特殊寄存器一览

附录

附录 1 外形尺寸图

- 附录 1.1 CPU 模块
- 附录 1.2 电源模块
- 附录 1.3 基板

附录 2 电池运输时的注意事项

索引

关于手册

导入手册

在构筑、设计安全系统之前，必须阅读以下手册。

手册名称	手册编号
安全应用指南 介绍安全系统的概要、安全系统的构筑方法、安装、连线以及应用程序等。 (另售)	SH-080716CHN

下列手册也与本产品有关。
请根据需要参考本表订购。

相关手册

手册名称	手册编号
QSCPU 用户手册 (硬件设计 / 维护点检篇) 介绍 QSCPU、安全电源模块以及安全基板等的规格。 (另售)	SH-080712CHN
QSCPU 编程手册 (公共指令篇) 介绍顺控程序指令、基本指令、应用指令以及 QSCPU 专用指令的使用方法有关内容。 (另售)	SH-080715CHN
CC-Link Safety 系统主站模块用户手册 (详细篇) QS0J61BT12 介绍 QS0J61BT12 型 CC-Link Safety 系统主站模块的规格、投运前的设置及步骤、参数设置以及故障排除有关内容。 (另售)	SH-080711CHN
CC-Link Safety 系统远程 I/O 模块用户手册 (详细篇) QS0J65BTB2-12DT 介绍 CC-Link Safety 系统远程 I/O 模块的规格、投运前的设置及步骤、参数设置以及故障排除有关内容。 (另售)	SH-080714CHN
Q 系列 MELSECNET/H 网络系统参考手册 (PLC 网络篇) 介绍 MELSECNET/H 网络系统的可编程控制器网络的规格、投运前的设置及步骤、参数设置以及故障排除有关内容。 (另售)	SH-080289C
GX Developer Version8 操作手册 介绍 GX Developer 中的编程方法、打印输出方法、监视方法以及调试方法等在线功能有关内容。 (另售)	SH-080311C
GX Developer Version8 操作手册 (安全可编程控制器篇) 介绍 GX Developer 功能中添加、更改的对应于安全可编程控制器的有关内容。 (另售)	SH-080576ENG

备注

准备有另售的印刷品，希望单独购买手册时，请通过上表中的手册编号购买。

手册的阅读方法

参照对象的表示

参照对象和参照手册用  标记来表示。

章节号的显示

由页右方的索引，翻开页所在的章节一目了然。

5 关于 CPU 模块中使用的存储器及文件

MELSEC Q Series

5.1.3 关于内置 ROM

- (1) 关于内置 ROM
内置 ROM 是用于在 CPU 模块中执行引导运行的存储器。
内置 ROM 可用于在无电池备份的状况下保存程序及参数等。
存储在内置 ROM 中的程序将通过引导（读取）到程序存储器（ 5.1.2 节）中进行运算。）
- (2) 可存储的数据
内置 ROM 中可以存储参数、程序、软元件注释。
关于各存储器中可存储的数据一览表，请参阅 5.1.1 节 (2)。
- (3) 存储器容量的确认
通过 GX Developer 的 [Online(在线)] → [Read from PLC(可编程控制器读取)] 可以确认存储器容量。
1) 在可编程控制器写入画面的对象存储器中，选择“Standard ROM(内置 ROM)”。

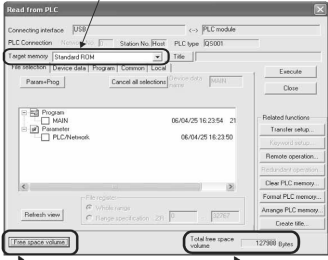


图 5.5 存储器容量的确认步骤

5.1 CPU 模块中使用的存储器
5.1.3 关于内置 ROM

节、项标题的显示

翻开页的节、项一目了然。

另外还有以下种类的说明。

☒ 要 点

介绍在相应页的内容中应特别注意的事项以及希望预先告知的功能等。

备 注

说明与相应页内容相关的参照目标以及预先告知会带来方便的内容。

本手册的使用方法

本手册是在用户使用 QS 系列可编程控制器时，帮助用户理解 CPU 模块中的存储映射、功能、程序、软元件等必备知识的一本手册。

本手册的构成大体可以分为以下几个部分：

- 1) 第 1 章 介绍 CPU 模块的概要。
- 2) 第 2 章至第 5 章 介绍 CPU 模块的性能规格、可执行程序、I/O 地址号、存储器等有关内容。
- 3) 第 6 章 介绍 CPU 模块功能的有关内容。
- 4) 第 7 章 介绍智能功能模块和存取方式有关内容。
- 5) 第 8 章至第 9 章 介绍 CPU 模块的参数、软元件有关内容。
- 6) 第 10 章 介绍 CPU 模块的处理时间有关内容。
- 7) 第 11 章 通过 GX Developer 将参数、程序写入到 CPU 模块中的步骤。

备注

在本手册中，未介绍电源模块、基板、扩展电缆、电池的规格等方面内容。
详细的情况请参阅下述手册：

 QSCPU 用户手册（硬件设计 / 维护点检篇）

本手册所使用的总称与略称

本手册中，除了特别说明的情况以外，使用如下所示的总称与略称来阐述关于 QS 系列 CPU 模块的有关内容。

总称 / 略称	总称 / 略称的内容
安全可编程控制器	安全 CPU 模块、安全电源模块、安全主基板、CC-Link Safety 主站模块、CC-Link Safety 远程 I/O 模块的总称。
常规可编程控制器	MELSEC-Q 系列、MELSEC-QnA 系列、MELSEC-A 系列、MELSEC-FX 系列的各种模块的总称。 (用于区别安全可编程控制器时)
QS 系列	三菱安全可编程控制器 MELSEC-QS 系列的略称。
QS001CPU	QS001CPU 型安全 CPU 模块的略称。
CPU 模块	QS001CPU 的别称。
GX Developer	产品型号 SW8D5C-GPPW、SW8D5C-GPPW-A、SW8D5C-GPPW-V、SW8D5C-GPPW-VA 的产品名的总称。
QS034B	QS034B 型安全主基板的略称。
基板	QS034B 的别称。
QS061P	QS061P-A1、QS061P-A2 型安全电源模块的略称。
电源模块	QS061P 的别称。
QS0J61BT12	QS0J61BT12 型 CC-Link Safety 系统主站模块的略称。
CC-Link Safety 主站模块	QS0J61BT12 的别称。
MELSECNET/H 模块	QJ71LP21-25、QJ71LP21S-25、QJ71LP21G、QJ71BR11 型 MELSECNET/H 网络模块的总称。
智能功能模块	CC-Link Safety 主站模块、MELSECNET/H 模块的总称。
QS0J65BTB2-12DT	QS0J65BTB2-12DT 型 CC-Link Safety 远程 I/O 模块的略称。
CC-Link Safety 远程 I/O 模块	QS0J65BTB2-12DT 的别称。
Q 系列 CPU 模块	Q00JCPU、Q00CPU、Q01CPU、Q02CPU、Q02HCPU、Q06HCPU、Q12HCPU、Q25HCPU、Q12PHCPU、Q25PHCPU、Q12PRHCPU、Q25PRHCPU 的总称。
常规 CPU 模块	Q 系列 CPU 模块的别称 (用于区别安全 CPU 模块时)
电池	Q6BAT 型电池的略称。
空槽盖	QG60 型空槽盖板的略称。
GOT	三菱图形操作终端 GOT-A*** 系列、GOT*** 系列、GOT1000 系列的总称。

备忘录

[illegible]

第 1 章 概要

本手册介绍了 QS 系列 CPU 模块 (QS001CPU) 的程序、I/O 地址号的分配方法、功能、软元件等有关内容。

关于电源模块、基板、电池的有关内容，请参阅以下手册。

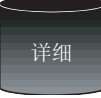
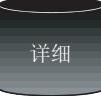
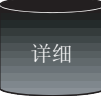
☞ QSCPU 用户手册 (硬件设计 / 维护点检篇)

(1) QS 系列 CPU 模块相关手册一览

以下手册中包含有与 QS 系列 CPU 模块相关的内容。

关于如下所示的手册编号的详细情况，请参阅本手册的“关于手册”。

表 1.1 QS 系列 CPU 模块相关手册一览

	 (附赠)	 维护点检篇	 程序基础篇	 公共指令篇
目的	QSCPU 型 CPU 模块用 户手册 (硬件篇)	QSCPU 用户手册 (硬件设计 / 维护点 检篇)	QSCPU 用户手册 (功能解说 / 程序基 础篇)	QSCPU 编程手册 (公共指令篇)
CPU 模块的部件名称和规格的确认	 概要	 详细	 概要	
电源模块、基板和 I/O 模块的连接方法的确认	 概要	 详细		
CPU 系统的构筑 (启动步骤和 I/O 地址号分配方法的确认)		 详细		
顺控程序组态和内存的确认			 详细	
CPU 模块的功能、参数和软元件的确认			 详细	
故障排除和出错代码的确认		 详细		
顺控指令、基本指令、应用指令等的使用方法的确认				 详细

1.1 特点

本节介绍 QS 系列 CPU 模块的特点。

(1) 可以构筑安全可编程控制器系统

QS 系列是作为可编程控制器所能取得的最高安全级别 ((IEC61508 SIL3、EN954-1/ISO13849-1 级别 4) 的可编程控制器。

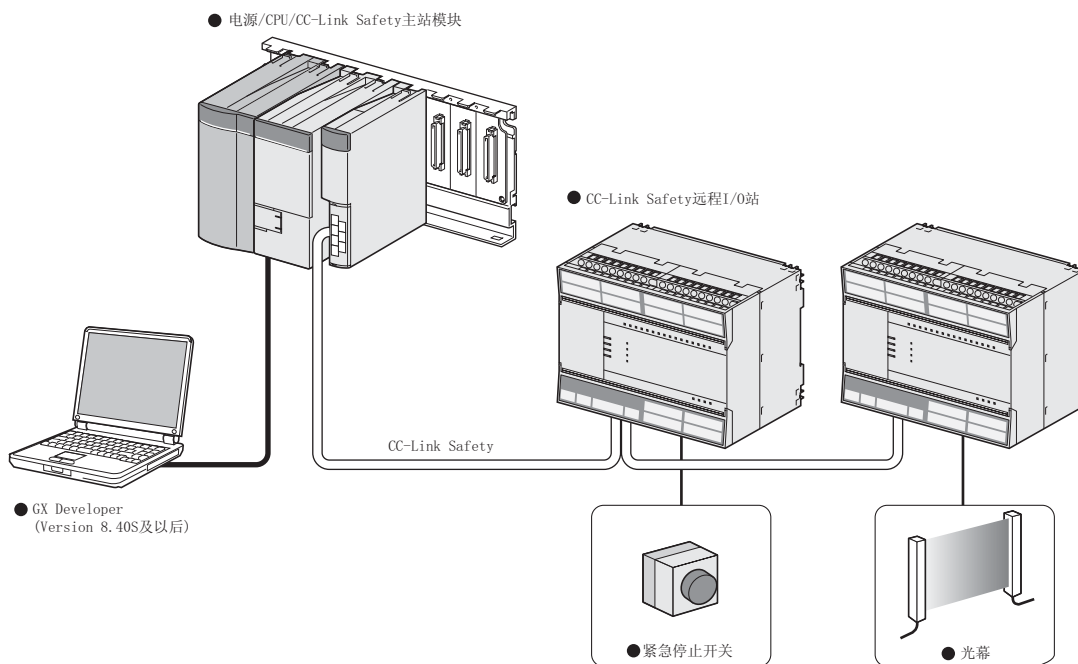


图 1.1 安全可编程控制器系统

(2) 配备了用于系统安全运行的安全 CPU 动作模式

CPU 模块配备了使系统安全运行的模式 “安全模式” 以及用于系统的构筑和维护的模式 “测试模式” 这 2 种安全 CPU 动作模式。

通过配备了这 2 种模式，可以防止用户的操作失误，保障系统的安全运行。

(a) 安全模式

安全模式是用于使系统安全运行的模式。在系统运行状态下禁止通过编程工具进行写入操作以及软元件测试操作等。

(b) 测试模式

测试模式是用于维护的模式。在该模式下可以通过编程工具进行写入操作及软元件测试操作等对顺控程序进行调试及维护。

(3) 详细的操作历史记录及故障历史记录

CPU 模块最多可以记录合计 3000 件的操作 / 故障历史记录，该历史记录对用户进行的 CPU 模块的操作内容及 CPU 模块和 CC-Link Safety 系统中发生的故障进行记录。通过将用户进行的 CPU 模块的操作内容作为操作 / 故障历史记录加以记录，在发生了故障时，可以通过操作 / 故障历史记录了解操作及故障的发生顺序。通过确认操作 / 故障历史记录，可以使故障排除易于进行。

操作 / 故障历史记录的记录内容如下表所示。

表 1.2 操作 / 故障历史记录

信息名称	内容	1 件历史记录信息
操作历史记录信息	将用户对 CPU 模块进行的操作作为历史记录加以保存。 (记录 CPU 模块的动作变更操作)	• 操作代码 • 操作信息 • 操作执行日期时间 • 结果代码 • 操作附属信息
故障历史记录信息	将以下故障作为历史记录保存。 • 自诊断检测出的故障、异常 • 硬件故障 • CC-Link Safety 系统中检测出的异常	• 出错代码 • 出错信息 • 发生日期时间 • 出错信息类型 (公共信息 / 个别信息) • 出错信息 (公共信息 / 个别信息)

(4) RAS 的强化

(a) 存储器诊断的强化

增强了 CPU 模块中配备的存储器的诊断功能。

(b) CPU 的冗余

配备了 2 个 CPU 模块 (CPU A 及 CPU B)。通过将 CPU A 及 CPU B 的运算结果进行校验, 在一致的情况下进行输出, 可以防止误输出。(校验结果不一致时系统将停运。)

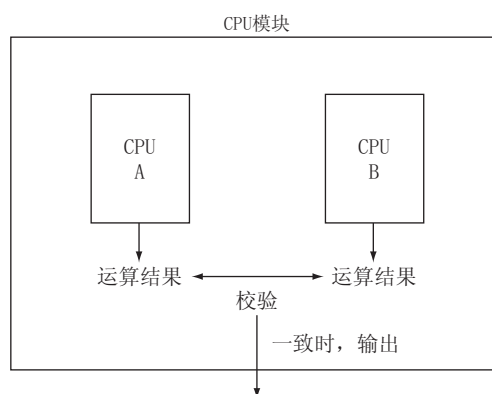


图 1.2 CPU 的冗余

(c) 通过硬件回路强化硬件诊断

通过表1.3的诊断功能, 在发生了无法通过OS检测的硬件异常时也可以防止误输出。

表 1.3 QS 系列 CPU 模块中新增的硬件诊断功能

诊断的名称	诊断内容
过电压、电压不足检测	检测电源模块向 CPU 模块供应的电源电压的过电压或电压不足。
时钟停止检测	检测至 CPU 模块内部回路的输入时钟的停止。

(5) 配备了 USB 接口

CPU 模块配备了作为与编程工具进行通信的手段的 USB 接口。

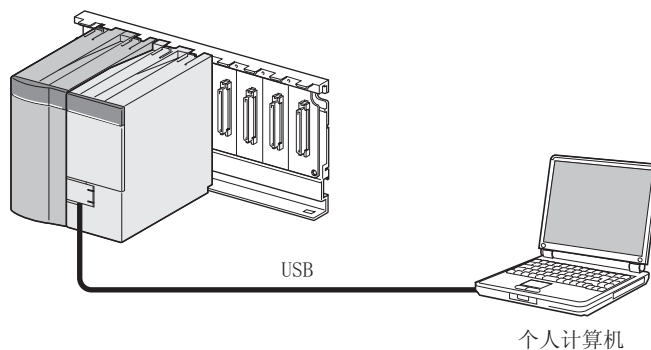


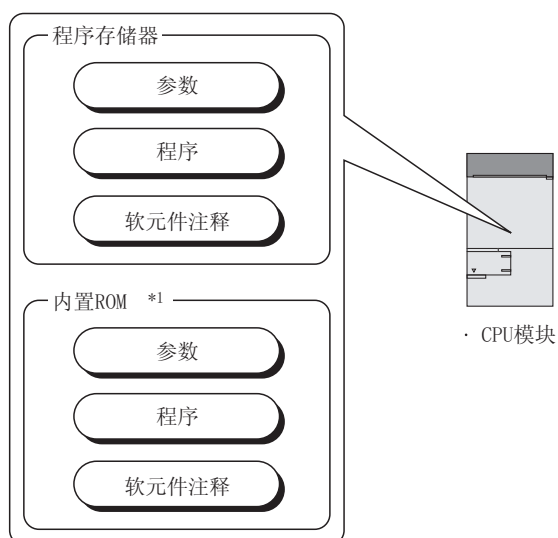
图 1.3 通过 USB 连接个人计算机

1.2 程序的存储及运算

(1) 程序的存储

(a) 存储通过 GX Developer 创建的程序

可以将通过 GX Developer 创建的程序存储到 CPU 模块的程序存储器、内置 ROM 中。



*1: 内置 ROM 用于对程序存储器进行 ROM 化。

图 1.4 存储器构成及存储目标

(b) 程序的执行

CPU 模块执行程序存储器中存储的程序的运算。

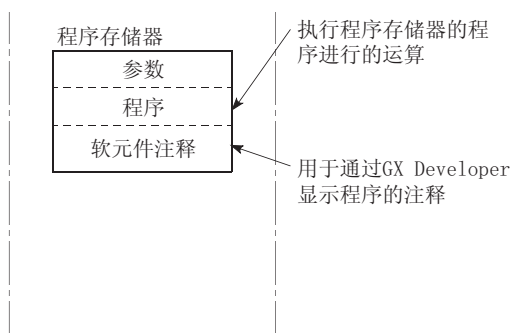


图 1.5 执行存储的程序

(c) 执行内置 ROM 中存储的程序

程序等也可以被存储到内置 ROM 中。

在进行了可编程控制器的电源 OFF → ON 或者 CPU 模块的复位解除时，存储在内置 ROM 中的程序可以被导入（读取）到程序存储器中执行。

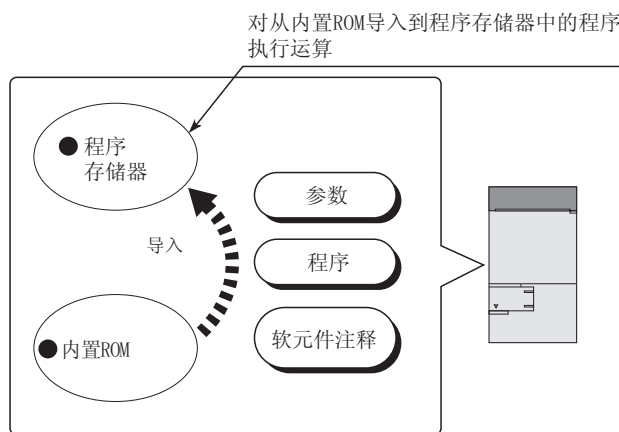


图 1.6 导入运行

1.3 便于编程的软元件、指令

CPU 模块中配备了便于创建程序的软元件、指令。
其大致概要如下所示。

(1) 可以进行灵活的软元件指定

在 CPU 模块中可以进行灵活的软元件指定。

(a) 将字软元件的各个位作为触点 - 线圈使用

通过字软元件的位指定，可以将字软元件的各个位作为触点 - 线圈使用。

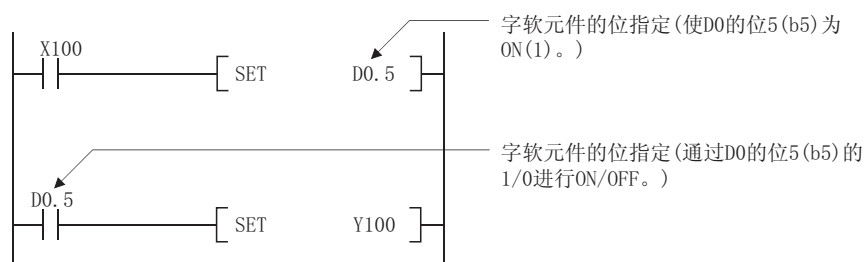


图 1.7 字软元件的位指定

(b) 通过使用微分触点，无需进行输入的脉冲化处理

通过使用微分触点 ($\neg$ / $\neg$), 不需要进行输入的脉冲化处理。

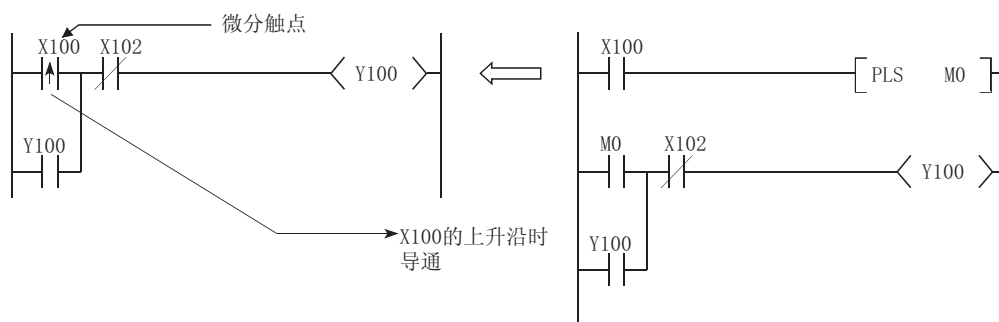


图 1.8 微分触点的使用

☒ 要 点

额定铭牌上记载的序列号与 GX Developer 的产品信息中显示的序列号有时会不一样。

- 额定铭牌的序列号表示产品的管理信息。
 - GX Developer 的产品信息中显示的序列号表示产品的功能信息。
产品的功能信息在新增功能时将被更新。
-

第 2 章 性能规格

CPU 模块的性能规格如下所示。

表 2.1 性能规格

项目		QS001CPU	备注
控制方式		反复执行存储程序运算	----
I/O 控制方式		刷新方式 ^{*3}	----
程序语言	程序控制语言	继电器符号语言、功能块	----
处理速度 (顺控程序指令)	LD X0	0.10μs	----
	MOV D0 D1	0.35μs	----
恒定扫描 (使扫描时间固定的功能)		1 ~ 2000ms (设置单位 :1ms)	通过参数设置
程序容量 ^{*1}		14k 步 (56k 字节)	☞ 5.1.1 节 5.1.2 节
存储器容量 ^{*1}	程序存储器 (驱动器 0)	128k 字节	☞ 5.1.2 节
	内置 ROM (驱动器 4)	128k 字节	☞ 5.1.3 节
最大存储文件个数	程序存储器	3 ^{*2}	☞ 5.1.2 节
	内置 ROM	3 ^{*2}	☞ 5.1.3 节
内置 ROM 的写入次数		最多 10 万次	----
I/O 软元件点数		6144 点 (X/Y0 ~ 17FF)	程序中可使用的点数
I/O 点数		1024 点 (X/Y0 ~ 3FF)	实际可存取 I/O 模块的 点数

*1 : 可执行的最大顺控程序步数的计算公式如下所示。
(程序容量) - (文件标题大小 (默认 :34 步))

☞ 关于程序容量、文件的详细内容, 请参阅第 5 章。

*2 : 可以将参数、顺控程序、软元件注释各存储为 1 个文件。

*3 : 刷新方式为: 在顺控程序的运算开始之前批量地进行 I/O 模块的存取的方式。

表 2.1 性能规格 (续)

项目		QS001CPU	备注
软元件点数	内部继电器 [M]	默认 6144 点 (M0 ~ 6143) (可更改)	使用点数可以在设置范围内更改 ➡ 9.2 节
	链接继电器 [B]	默认 2048 点 (B0 ~ 7FF) (可更改)	
	定时器 [T]	默认 512 点 (T0 ~ 511) (可更改) (低速定时器 / 高速定时器共用)	
		通过指令指定低速定时器 / 高速定时器 低速定时器 / 高速定时器的计量单位是在参数中设置 (低速定时器 : 1 ~ 1000ms, 单位 1ms, 默认 100ms) (高速定时器 : 0.1 ~ 100ms, 单位 0.1ms, 默认 10ms)	
		默认 0 点 (低速定时器 / 高速定时器共用) (可更改) 通过指令指定低速定时器 / 高速定时器 低速定时器 / 高速定时器的计量单位是在参数中设置 (低速定时器 : 1 ~ 1000ms, 单位 1ms, 默认 100ms) (高速定时器 : 0.1 ~ 100ms, 单位 0.1ms, 默认 10ms)	
	累计定时器 [ST]	默认 0 点 (低速定时器 / 高速定时器共用) (可更改) 通过指令指定低速定时器 / 高速定时器 低速定时器 / 高速定时器的计量单位是在参数中设置 (低速定时器 : 1 ~ 1000ms, 单位 1ms, 默认 100ms) (高速定时器 : 0.1 ~ 100ms, 单位 0.1ms, 默认 10ms)	
	计数器 [C]	• 普通计数器 : 默认 512 点 (C0 ~ 511) (可更改)	
	数据寄存器 [D]	默认 6144 点 (D0 ~ 7FF6143) (可更改)	
	链接寄存器 [W]	默认 2048 点 (W0 ~ 7FF) (可更改)	
	报警器 [F]	默认 1024 点 (F0 ~ 1023) (可更改)	
	变址继电器 [V]	默认 1024 点 (V0 ~ 1023) (可更改)	
	链接特殊继电器 [SB]	1536 点 (SB0 ~ 5FF)	软元件点数固定
	链接特殊寄存器 [SW]	1536 点 (SW0 ~ 5FF)	
	特殊继电器 [SM]	5120 点 (SM0 ~ 5119)	
	特殊寄存器 [SD]	5120 点 (SD0 ~ 5119)	
RUN/PAUSE 触点		根据 X0 ~ 17FF 可设置 1 个 RUN 触点, 无 PAUSE 触点。	根据参数设置
时钟功能		年、月、日、时、分、秒、星期 (自动判断闰年) 精度 : -3.18 ~ +5.25s (TYP. +2.14s) / d 在 0℃ 环境下 精度 : -3.18 ~ +2.59s (TYP. +2.07s) / d 在 25℃ 环境下 精度 : -12.97 ~ +3.63s (TYP. -3.16s) / d 在 55℃ 环境下	➡ 6.11 节
允许掉电时间		根据电源模块	----
DC5V 内部消耗电流		0.43A	----
外形尺寸	H	98mm	----
	W	55.2mm	----
	D	115mm	----
重量		0.29kg	----
保护级别		IP2X	----

备注

关于一般规格的有关内容请参阅以下手册。

➡ QSCPU 用户手册 (硬件设计 / 维护点检篇)

第 3 章 顺控程序的执行

CPU 模块按图 3.1 中所示的步骤执行程序。

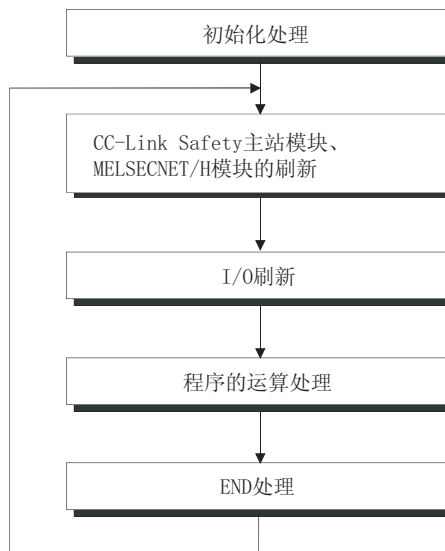


图 3.1 程序执行步骤

1

概要

2

性能规格

3

顺控程序的执行

4

I/O 地址号的分配

5

关于 CPU 模块中使用的存
储器及文件

6

功能

7

与智能功能模块的通信

8

参数

3.1 顺控程序

顺控程序是使用顺控程序指令、基本指令、应用指令等创建的程序。

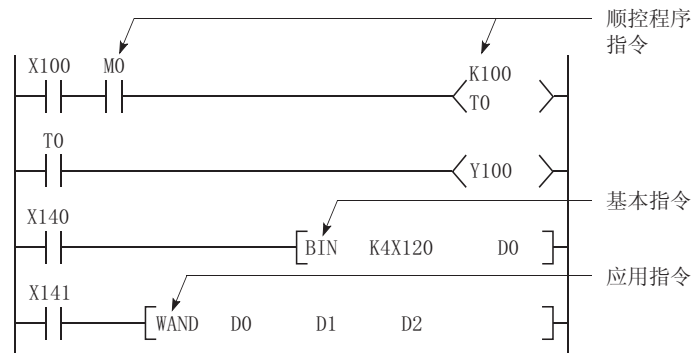


图 3.2 顺控程序

备注

关于顺控程序指令、基本指令、应用指令，请参阅以下手册。

☞ QSCPU 编程手册（公共指令篇）

3.1.1 顺控程序的记述方法

顺控程序是以 GX Developer 的梯形图模式创建。

梯形图模式是基于继电器控制的顺控程序电路的思路的模式。可以以类似于顺控程序电路的方式编程。

梯形图模式的编程是以梯形图块为单位进行的。

梯形图块是执行顺控程序运算的最小单位。

梯形图块是从左侧的纵母线开始至右侧的纵母线为止的电路。

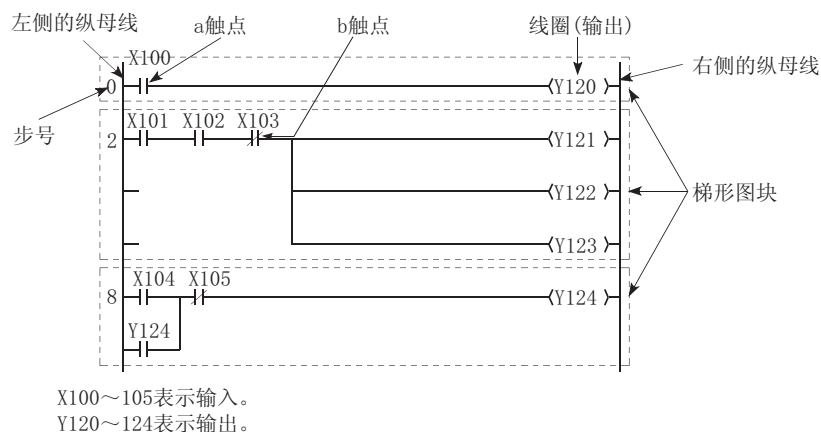


图 3.3 梯形图模式

3.1.2 顺控程序的运算

(1) 梯形图块的运算顺序

CPU 模块是从左侧的纵母线开始按从左至右，从上至下的顺序执行运算。

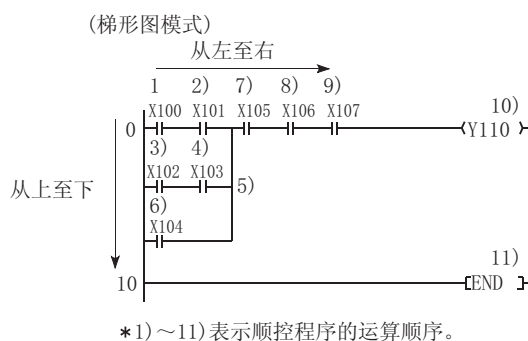


图 3.4 梯形图块的运算顺序

(2) 顺控程序的执行动作

顺控程序从步 0 开始执行 END 命令，进行 END 处理。

END 处理后再次从步 0 开始运算。

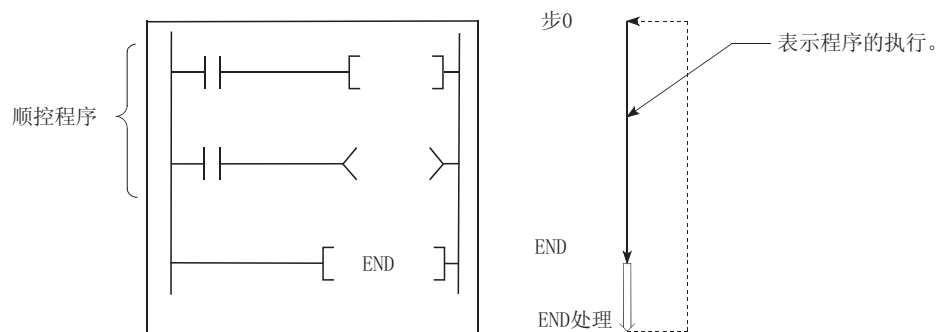


图 3.5 顺控程序

3.2 扫描时间的思路

(1) 扫描时间

是 CPU 模块从顺控程序的步 0 开始执行运算，至再次执行同一顺控程序的步 0 为止所耗费的时间。

扫描时间是由顺控程序的执行时间以及 END 处理时间所构成。

(a) 扫描时间的存储位置

在 CPU 模块中，计量扫描时间的当前值、最小值、最大值，并存储到特殊寄存器 (SD520、SD521、SD524 ~ 527) 中。

通过监视 SD520、SD521、SD524 ~ 527，可以确认扫描时间。

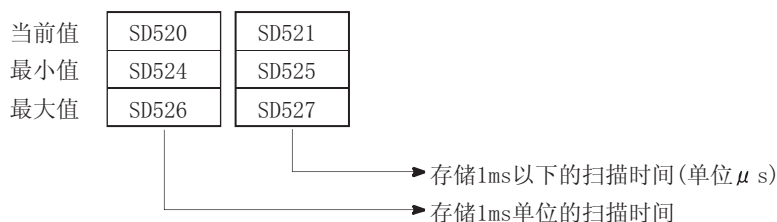


图 3.6 扫描时间的存储位置

SD520 为 3、SD521 为 400 时，扫描时间变为 3.4ms。

(b) 扫描时间的精度及计量

特殊寄存器中存储的各扫描时间的精度为 $\pm 0.1ms$ 。

(c) 扫描时间的监视

CPU 模块中配备有监视扫描时间的定时器 (看门狗定时器)。(本节 (2))

(2) WDT (看门狗定时器)

看门狗定时器 (以下略称为 WDT) 是监视扫描时间的定时器。

默认值被设置为 200ms。

(a) WDT 的误差

WDT 的误差为 10ms。

如果将 WDT(t) 设置为 10ms，扫描时间在 $10ms < t < 20ms$ 的范围内将变为 “WDT ERROR”。

(b) WDT 的设置

WDT 在可编程控制器参数的 PC RAS 设置中可以在 10ms ~ 2000ms 的范围内更改。

(设置单位 : 10ms)

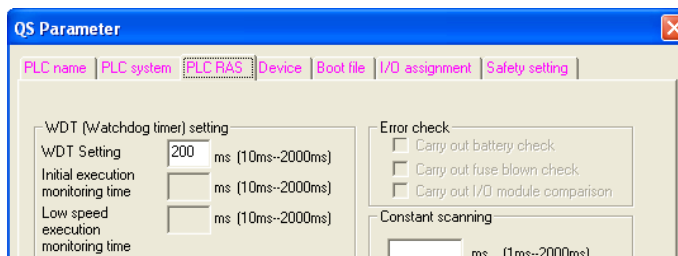


图 3.7 PC RAS 设置 (WDT 设置)

(3) 以一定的间隔反复执行程序的功能

使用恒定扫描时间功能 ( 6.9 节) 可以以一定的间隔反复执行程序。
进行了恒定扫描设置时，将在所设置的各恒定扫描时间内执行程序。

3.3 运算处理

本节介绍 CPU 模块的运算处理有关内容。

3.3.1 初始化处理

初始化处理是用于执行顺控程序的运算的前处理。

在接通可编程控制器的电源时或进行 CPU 模块的复位解除时仅执行一次以下处理。

- 系统设置
- 来自于内置 ROM 的引导 *
- 安全 CPU 动作模式设置
- 自诊断
- CC-Link Safety 网络信息的设置
- MELSECNET/H 网络信息的设置
- CPU 动作状态确定

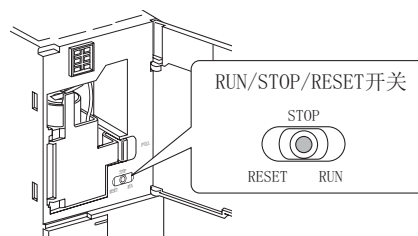
初始化处理结束后，CPU 模块将进入 RUN/STOP/RESET 开关设置中设置的动作状态。

( 3.4 节)

*: 安全模式时，与可编程控制器参数的引导文件设置无关，执行来自于内置 ROM 的引导。
测试模式的情况下，在可编程控制器参数的引导文件设置中设置了从内置 ROM 引导时，
执行来自于内置 ROM 的引导。

☒ 要 点

1. CPU 模块的 RUN/STOP/RESET 开关如下图所示。



2. 在 STOP 状态下更改了参数或者程序时，应通过 RUN/STOP/RESET 开关进行复位。

3.3.2 CC-Link Safety、MELSECNET/H 的刷新

执行在网络参数的 CC-Link Safety 及 MELSECNET/H 中设置的刷新范围的刷新。

CC-Link Safety、MELSECNET/H 的刷新是在顺控程序的运算开始之前执行。

执行恒定扫描时，在各恒定扫描时间内执行 CC-Link Safety、MELSECNET/H 的刷新。

3.3.3 I/O 刷新

在 I/O 刷新中，执行与 CC-Link Safety 主站模块、MELSECNET/H 模块的 I/O 数据的刷新。
I/O 刷新是在 CC-Link Safety、MELSECNET/H 的刷新之后执行。

3.3.4 END 处理

END 处理是用于结束 1 个扫描的顺控程序运算处理，使顺控程序的执行返回至步 0 的后处理。

END 处理中包含了以下处理。

- 自诊断处理 (☞ 6.7 节)
- 与 GX Developer 通信
- 看门狗定时器的复位 (☞ 6.15 节)
- 设置的时机为 END 处理时的至特殊继电器 / 特殊寄存器的值的设置 (☞ 附录 1、☞ 附录 2)
- 等待恒定扫描时间

☒ 要 点

在设置了恒定扫描功能 (☞ 6.9 节) 的情况下，在 END 处理后或下一个扫描开始之前的期间，END 处理时的结果将被保持。

3.4 RUN 状态、STOP 状态的运算处理

CPU 模块的动作状态分为 RUN 状态及 STOP 状态这 2 种。
以下介绍在各动作状态下的 CPU 模块的运算处理。

(1) RUN 状态的运算处理

RUN 状态是指，顺控程序的运算按步 0 → END 命令 → 步 0 的顺序反复执行的状态。

(a) 进入 RUN 状态时的输出状态

进入 RUN 状态时，根据参数的 STOP → RUN 时的输出模式设置，输出 STOP 状态时退避的输出 (Y) 的状态，或者输出 1 个扫描后的运算结果。(6.10 节)

(b) 至运算开始前的处理时间

进行了 STOP → RUN 的切换后，至顺控程序运算开始前的处理时间依据系统配置及参数设置内容而变动。(通常为 0.1 秒)

(2) STOP 状态的运算处理

STOP 状态是指，通过 RUN/STOP/RESET 开关或者远程 STOP 功能 (6.12.1 节)，终止顺控程序的运算的状态。

此外，在发生了停止出错的情况下也将变为 STOP 状态。

(a) 进入 STOP 状态时的输出状态

进入 STOP 状态时，输出 (Y) 的状态将退避，输出全部被 OFF。
除输出 (Y) 以外的软元件存储器将被保持。

(3) 开关操作时的 CPU 模块的运算处理

表 3.1 开关操作时的运算处理

RUN/STOP 状态	CPU 模块的运算处理			
	顺控程序的运算处理	外部输出	软元件存储器	
			M、T、C、D	Y
RUN→STOP	执行至 END 命令时为止。	变为 STOP 状态之前的输出 (Y) 的状态将退避，输出全部被 OFF。	保持变为 STOP 状态之前的软元件存储器的状态。	变为 STOP 状态之前的输出 (Y) 的状态将退避，输出全部被 OFF。
STOP→RUN	从步 0 开始。	取决于可编程控制器参数的“STOP→RUN 时的输出模式”。	使用 STOP 状态的软元件存储器的状态。	取决于可编程控制器参数的“STOP→RUN 时的输出模式”。 ( 6.10 节)

☒ 要 点

CPU 模块无论是在 RUN 状态还是 STOP 状态下均执行以下处理。

- I/O 刷新处理
- CC-Link Safety 主站模块的自动刷新处理
- MELSECNET/H 模块的刷新处理
- 自诊断处理
- 与 GX Developer 的通信处理

因此即使是在 STOP 状态下，也可以通过 GX Developer 进行输入输出的监视以及测试操作。

3.5 瞬间掉电时的运算处理

CPU 模块在电源模块的输入电源电压低于规定范围时将检测出瞬间掉电，并执行以下的运算处理。

(1) 发生了允许掉电时间以下的瞬间掉电时

发生了瞬间掉电时，保持输出状态不变，进行了出错历史记录的登录后，中断运算处理。

(定时器的计量将继续。)

(a) 瞬间掉电被解除时

瞬间掉电被解除时，继续执行运算处理。

(b) 发生瞬间掉电时的看门狗定时器 (WDT) 的计量

即使发生了瞬间掉电，中断了运算之后，看门狗定时器 (WDT) 的计量仍将继续。

例如，可编程控制器参数的 WDT 设置为 200ms，扫描时间为 190ms 时，如果发生了 15ms 的瞬间掉电，将发生看门狗定时器出错。

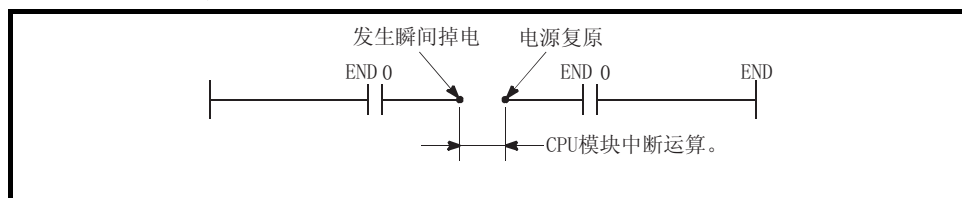


图 3.8 发生瞬间掉电时的运算处理

(2) 发生了超出允许掉电时间的掉电时

CPU 模块将开始初始化。

将执行与进行了以下操作时相同的运算处理。

- 接通可编程控制器的电源
- 通过 RUN/STOP/RESET 开关进行复位操作
- 通过 GX Developer 进行远程复位操作

3.6 数据的清除处理

本节介绍 CPU 模块的数据清除方法。

(1) 数据的清除方法

CPU 模块的数据清除方法有如下 6 种。

- (a) 通过 RUN/STOP/RESET 开关、GX Developer 进行复位操作
- (b) 重启可编程控制器系统的电源
- (c) 通过 GX Developer 进行可编程控制器存储器清除
- (d) 通过 GX Developer 进行可编程控制器存储器格式化
- (e) 通过 GX Developer 进行可编程控制器存储器的初始化
- (f) 通过 GX Developer 进行历史记录清除 (操作 / 故障历史记录清除)

(2) 通过各清除方法进行数据清除的可否

通过 (1) 的 (a) ~ (f) 中所示的清除方法进行数据清除的可否如表 3.2 所示。

表 3.2 通过各清除方法进行数据清除的可否

数据项目	数据的清除方法					
	复位操作	电源的重启	可编程控制器存储器清除	可编程控制器存储器格式化	可编程控制器存储器初始化	历史记录清除
程序存储器的数据	×	×	×	○	○	×
内置 ROM 的数据 *1	×	×	×	×	○	×
软元件数据	○	○	○	×	○	×
安全 CPOU 动作模式	×	×	×	×	○*2	×
CPU 访问密码	×	×	×	×	○	×
时钟数据	×	×	×	×	○	×
操作 / 故障历史记录	×	×	×	×	○*3	○*4
ROM 写入次数	×	×	×	×	×	×

○ : 可清除的数据 × : 无法清除的数据

*1 : 通过 GX Developer 进行程序存储器的 ROM 化时, 将内置 ROM 的数据清除之后再执行至内置 ROM 的写入。

*2 : 如果执行可编程控制器存储器的初始化, 安全 CPU 动作模式将变为测试模式。

*3 : 删除历史记录后, 下述可编程控制器存储器初始化的操作 / 故障历史记录将被记录。

- OP005 : FSYSTEM INITIALIZE PLC MEMORY
- OP100 : POWER ON
- 2200 : MISSING PARAMETER

*4 : 删除了操作 / 故障历史记录后, 下述操作历史记录将被记录。

- OP200 : CLEAR OPERATION/ERROR LOG

备注

关于 GX Developer 的操作方法, 请参阅以下手册。

☞ GX Developer Version 8 操作手册

☞ GX Developer Version 8 操作手册 (安全可编程控制器篇)

3.7 顺控程序中可使用的数值

在 CPU 模块中，将数值、字母等的的数据以 0(OFF) 及 1(ON) 的 2 个状态表示。

将以 0 和 1 表示的数据称为 BIN(2 进制数)。

在 CPU 模块中，也可以使用将 BIN 数据以 4 位为单位表示的 HEX(16 进制数)、BCD(2 进制编码的 10 进制数)。

BIN、HEX、BCD、DEC(10 进制数) 的数值表示如下表 3.3 所示。

表 3.3 BIN、HEX、BCD、DEC 的数值表示

DEC(10 进制数)	HEX(16 进制数)	BIN(2 进制数)				BCD(2 进制编码的 10 进制数)			
0	0				0				0
1	1				1				1
2	2				10				10
3	3				11				11
.	.				.				.
.	.				.				.
.	.				.				.
9	9				1001				1001
10	A				1010			1	0000
11	B				1011			1	0001
12	C				1100			1	0010
13	D				1101			1	0011
14	E				1110			1	0100
15	F				1111			1	0101
16	10		1		0000			1	0110
17	11		1		0001			1	0111
.	.				.				.
.	.				.				.
.	.				.				.
47	2F		10		1111			100	0111
.	.								
.	.								
.	.								
32766	7FFE	0111	1111	1111	1110		--		
32767	7FFF	0111	1111	1111	1111		--		
-32768	8000	1000	0000	0000	0000	1000	0000	0000	0000
-32767	8001	1000	0000	0000	0001	1000	0000	0000	0001
.	.								
.	.								
.	.								
-2	FFFE	1111	1111	1111	1110		--		
-1	FFFF	1111	1111	1111	1111		--		

(1) 从外部至 CPU 模块的数值输入

从外部通过数字开关等将数值设置到 CPU 模块中时，通过下述 (b) 中所示的方法，可以通过 BCD(2 进制编码的 10 进制数) 进行与 DEC(10 进制数) 相同的设置。

(a) CPU 模块内部中使用的数值

CPU 模块是以 BIN(2 进制数) 进行运算的。

如果原样不变地使用以 BCD 设置的值，CPU 模块将把设置的值作为 BIN 进行运算。因此，将以与所设置的值相异的值进行运算。

(☞ 下述 (b))

(b) 无需在意 BIN 的数值输入方法

将以 BCD 设置的数据转换为 CPU 模块中使用的 BIN 数据时，使用 BIN 命令。

如果使用 BIN 命令，可以在无需在意 BIN 的状况下从外部进行数值数据设置。

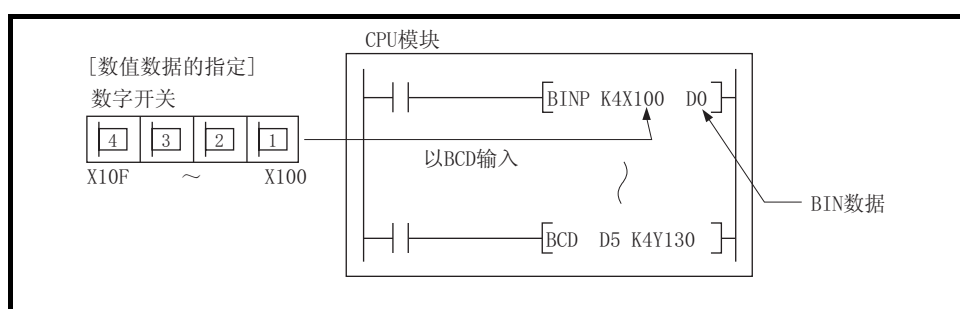


图 3.9 从数字开关至 CPU 模块的数据获取

备注

关于 BIN 命令的详细内容，请参阅以下手册。

☞ QSCPU 编程手册 (公共指令篇)

(2) 从 CPU 模块至外部的数值输出

将 CPU 模块中运算的数值显示到外部时，可以使用数字显示器等。

(a) 数值输出方法

CPU 模块以 BIN 进行运算。

将 CPU 模块中使用的 BIN 原样不变地输出到数字显示器时，将无法正确显示。

为了将 BIN 运算的数据转换为外部显示器等中使用的 BCD，应使用 BCD 命令。

通过使用 BCD 命令，可以在外部显示器等中显示与 DEC(10 进制数) 相同的显示。

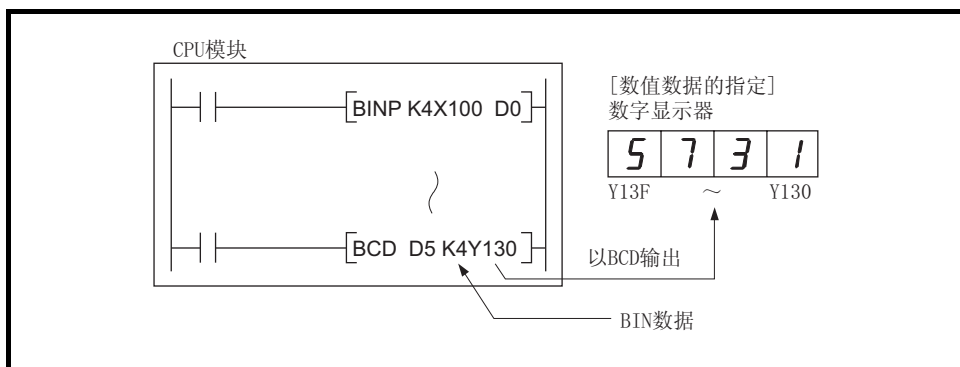


图 3.10 通过数字显示器显示 CPU 模块的运算数据

备注

关于 BCD 命令的详细内容，请参阅以下手册。

☞ QSCPU 编程手册（公共指令篇）

3.7.1 BIN(2 进制数 : Binary Code)

(1) 2 进制数

BIN(2 进制数) 是以 0(OFF) 及 1(ON) 表示的数值。

在 DEC(10 进制数) 中从 0 增加至 9 时, 下一个进位将变为 10。

在 BIN 中 0、1 的下一个进位将变为 10(10 进制数的 2)。

BIN 及 DEC 的数值表示如表 3.4 所示。

表 3.4 BIN 与 DEC 的数值表示的区别

DEC(10 进制数)	BIN(2 进制数)	
0	0000	
1	0001	
2	0010	← 进位
3	0011	
4	0100	← 进位
5	0101	
6	0110	
7	0111	
8	1000	← 进位
9	1001	
10	1010	
11	1011	

(2) BIN 的数值表示

(a) CPU 模块中使用的 BIN 的位构成

CPU 模块的各寄存器 (数据寄存器、链接寄存器等) 是由 16 位所构成。

(b) CPU 模块中可使用的数值数据

CPU 模块的各寄存器中可以存储 -32768 ~ 32767 的数值。

CPU 模块的各寄存器的数值表示如图 3.11 所示。

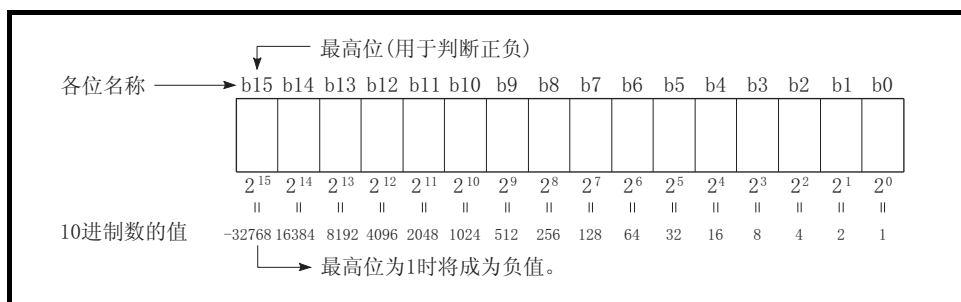


图 3.11 CPU 模块的各寄存器的数值表示

☒ 要 点

各寄存器的各位中，分配了 2^n 的数值。

但是，最高位的位是用来判别正负的，因此不能使用不带有符号（无符号）的 BIN。

- 1) 最高位为 0... 正
- 2) 最高位为 1... 负

1

概要

2

性能规格

3

顺控程序的执行

4

I/O 地址号的分配

5

关于 CPU 模块中使用的存储器及文件

6

功能

7

与智能功能模块的通信

8

参数

3.7.2 HEX(16 进制数 :Hexadecimal)

(1) HEX

HEX(16 进制数) 是将 BIN 数据的 4 位作为 1 位表示的。

如果在 BIN(2 进制数) 中使用 4 位, 可以表示 0 ~ 15 的 16 个数值。

在 HEX 中, 0 ~ 15 是用 1 位表示, 因此将 9 的下一位 10 用字母 AH, 将 11 用字母 BH 表示, 在 FH(15) 的下一位进位

BIN、HEX、DEC(10 进制数) 的数值表示如表 3.5 所示。

表 3.5 BIN、HEX、DEC 的数值表示

DEC(10 进制数)	HEX(16 进制数)	BIN(2 进制数)	
0	0		0
1	1		1
2	2		10
3	3		11
.	.		.
.	.		.
.	.		.
9	9		1001
10	A		1010
11	B		1011
12	C		1100
13	D		1101
14	E		1110
15	F		1111
16	10	1	0000
17	11	1	0001
.	.		.
.	.		.
.	.		.
47	2F	10	1111

进位

(2) HEX 的数值表示

CPU 模块的各寄存器 (数据寄存器、链接寄存器等) 是由 16 位所构成。

16 位的情况下, 在 16 进制数中可以指定 0 ~ FFFF_H。

☒ 要 点

各寄存器均使用带符号的数值。

可指定的数值范围与 BIN 相同, 为 -32768 ~ 32767。

3.7.3 BCD(2 进制编码的 10 进制数 :Binary Coded Decimal)

(1) BCD

BCD(2 进制编码的 10 进制数) 是将 DEC(10 进制数) 的 1 位以 BIN(2 进制数) 表示的数值。

与 HEX(16 进制数) 一样以 4 位表示, 但不使用 HEX 的 A_H ~ F_H。

BIN、BCD、DEC 的数值表示如表 3.6 所示。

表 3.6 BIN、BCD、DEC 的数值表示

DEC(10 进制数)	BIN(2 进制数)	BCD (2 进制编码的 10 进制数)	
0	0000	0	
1	0001	1	
2	0010	10	
3	0011	11	
4	0100	100	
5	0101	101	
6	0110	110	
7	0111	111	
8	1000	1000	
9	1001	1001	
10	1010	1 0000	进位
11	1011	1 0001	
12	1100	1 0010	

(2) BCD 的数值表示

CPU 模块的各寄存器(数据寄存器、链接寄存器等)是由 16 位所构成。

因此, 将各寄存器中可存储的数值以 BCD 表示时, 其范围变为 0 ~ 9999。

备忘录

[illegible]

第 4 章 I/O 地址号的分配

本章介绍 CPU 模块与 I/O 模块、智能功能模块进行数据收发时所必需的 I/O 地址号的分配有关内容。

4.1 关于 I/O 地址号

I/O 地址号是指，通过顺控程序获取至 CPU 模块的 ON/OFF 数据时，以及从 CPU 模块向外部输出 ON/OFF 数据时使用的地址。

(1) ON/OFF 数据的获取及输出

获取至 CPU 模块的 ON/OFF 数据是通过输入 (X)，从 CPU 模块向外部输出 ON/OFF 数据是通过输出 (Y) 进行的。

(2) I/O 地址号的表示

I/O 地址号是用 16 进制数表示。

4.2 I/O 地址号的分配思路

4.2.1 基板的 I/O 地址号

接通 CPU 模块的电源或者进行复位解除时，进行 I/O 地址号的分配。
I/O 地址号是从主基板的 CPU 模块的右侧开始自动分配。

在主基板上安装了 2 个 CC-Link Safety 主站模块、1 个 MELSECNET/H 模块时的 I/O 地址号如图 4.1 所示。

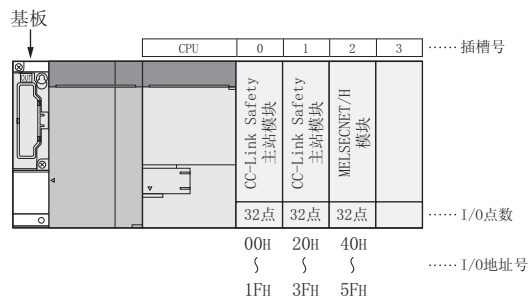


图 4.1 I/O 地址号的分配示例

对于在主基板上未安装 CC-Link Safety 主站模块、MELSECNET/H 模块的空插槽，将分配在可编程控制器参数的可编程控制器系统设置中设置的点数。（默认为 16 点。）

备注

在 GX Developer 的 I/O 分配中可以更改各插槽的起始 I/O。

4.2.2 远程站的 I/O 地址号

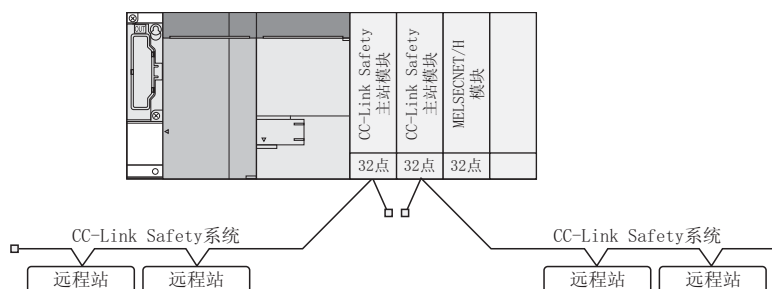
在 CC-Link Safety 系统中，可以对远程站的 I/O 模块分配 CPU 模块的输入 (X)、输出 (Y) 并进行控制。

(1) 远程站中可使用的 CPU 模块的 I/O 地址号

在主基板上安装了 2 个 CC-Link Safety 主站模块、1 个 MELSECNET/H 模块时，使用 X/Y0 ~ X/Y5F。

远程站的 I/O 地址号使用 CPU 模块的输入 (X)、输出 (Y) 时，应使用 X/Y60 或以后。

[系统配置]



[I/O 地址号的分配]

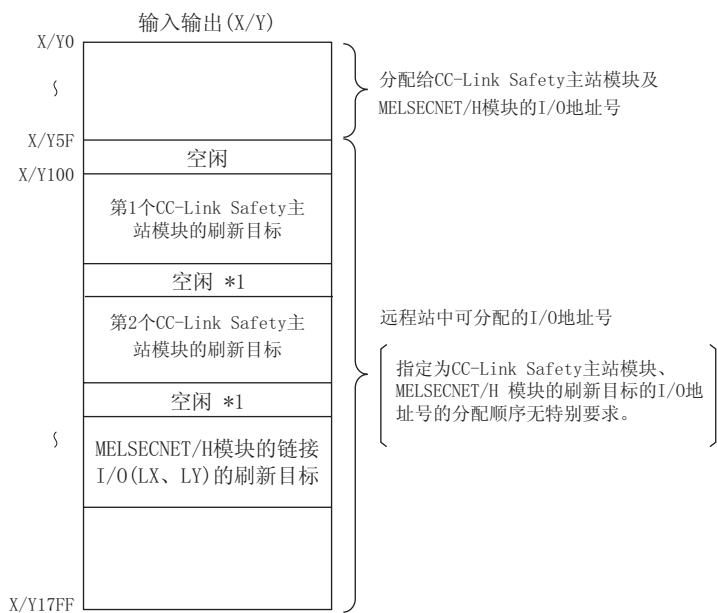


图 4.2 远程站的 I/O 地址号分配

备注

在以下的刷新目标之间也可设置空位。

- 在第 1 个与第 2 个 CC-Link Safety 主站模块的刷新目标之间
- 在第 2 个 CC-Link Safety 主站模块与 MELSECNET/H 模块的链接 I/O 的刷新目标之间

☒ 要 点

1. CPU 模块的输入 (X)、输出 (Y) 可以被用于 MELSECNET/H 网络模块的链接 I/O(LX、LY) 的刷新目标 (CPU 模块侧软元件)。
 2. 使用了多个 CC-Link Safety 主站模块时, 注意应避免刷新目标的 I/O 地址号重复。
 3. 使用了 CC-Link Safety 主站模块、MELSECNET/H 模块时, 注意应避免刷新目标的 I/O 地址号重复。
-

4.3 通过 GX Developer 进行 I/O 分配

本节介绍通过 GX Developer 进行 I/O 分配的有关内容。

4.3.1 通过 GX Developer 进行 I/O 分配的目的

在以下的情况下通过 GX Developer 进行 I/O 分配设置。

- (1) 防止更换模块时 I/O 地址号发生变化
可以防止由于智能功能模块故障而卸下模块时 I/O 地址号发生变化。
- (2) 更改程序中使用的 I/O 地址号
所设计的程序中使用的 I/O 地址号与实际系统的 I/O 地址号不一致时，可以将基板的各模块的 I/O 地址号更改为程序的 I/O 地址号。

☒ 要 点

1. I/O 分配设置在可编程控制器的电源 OFF → ON 或者 CPU 模块的复位解除时生效。
2. 在未通过 GX Developer 进行 I/O 分配的状态下，当智能功能模块故障时，该模块后面的 I/O 地址号将发生变化，有可能导致误动作。
因此，建议通过 GX Developer 进行 I/O 分配设置。

4.3.2 通过 GX Developer 进行 I/O 分配的思路

在 I/O 分配中，可以对基板的各插槽分别设置“类型”（模块类型）、“点数”（I/O 点数）、“起始 XY”（起始 I/O 地址号）。

例如，更改指定插槽的 I/O 点数时，可以仅对点数进行设置。

未设置的项目将保持基板的安装状态不变。

(1) I/O 分配的设置

I/O 分配是在可编程控制器参数的 I/O 分配设置中进行。

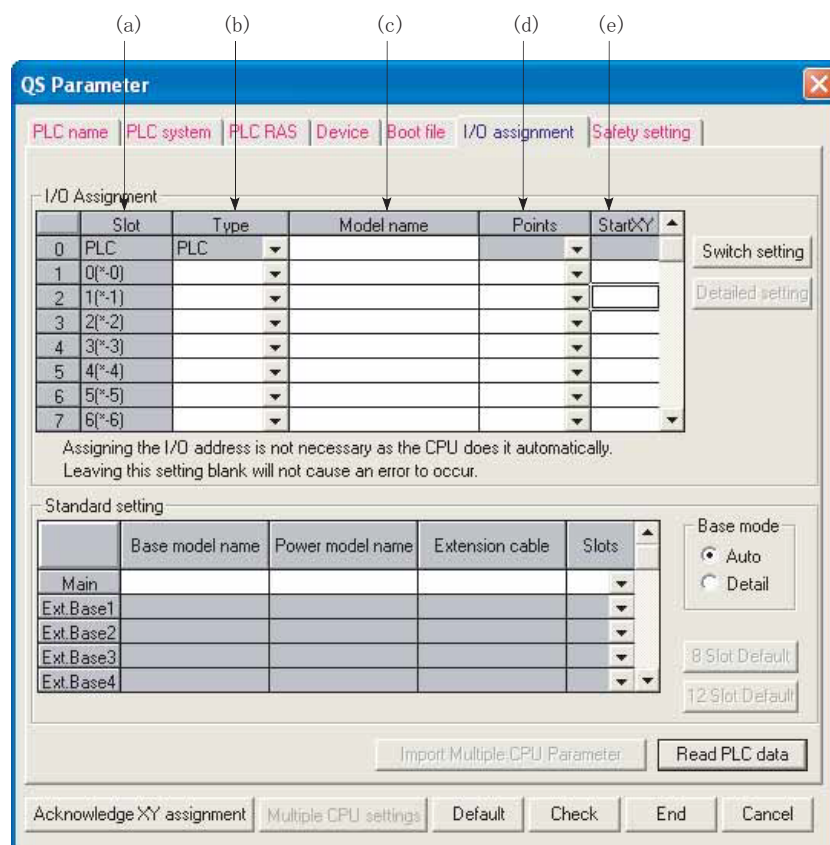


图 4.3 I/O 分配设置

(a) 插槽

显示插槽号及位于主基板的第几个插槽。

位于第几个插槽即为主基板的 0 插槽开始的插槽数。

(b) 类型

将安装了 CC-Link Safety 主站模块、MELSECNET/H 模块的插槽选择为智能。
将空插槽选择为空闲。
未设置类型的插槽将变为实际安装的模块类型。

(c) 型号

在半角 16 个字符以内设置安装的模块型号。
CPU 模块中不使用所设置的型号。(仅作为用户的备忘。)

(d) 点数

更改各插槽的 I/O 点数时，应从以下点数中选择。

- | | | |
|---------------|---------------|-------------------|
| · 0(0 点) | · 16(16 点) | · 32(32 点) |
| · 48(48 点) | · 64(64 点) | · 128(128 点) |
| · 256(256 点) | · 512(512 点) | · 点 1024(1024 点) |

对于未设置点数的插槽，将成为实际安装的模块的点数。

(e) 起始 XY

更改各插槽的 I/O 地址号时，设置更改以后的起始 I/O 地址号。
对于未设置起始 XY 的插槽，将从所设置的插槽开始以连号进行 I/O 地址号的分配。

(2) I/O 分配时的注意事项

(a) 进行了 I/O 分配的插槽状态

对于进行了 I/O 分配的插槽，与模块的实际安装无关，I/O 分配设置将优先。

1) 设置的点数少于实际安装的智能功能模块的点数时
将发生“MODULE LAYOUT ERROR”。

2) 实际安装的模块与 I/O 分配的类型

应使 I/O 分配设置的类型与实际安装的模块相符合。

I/O 分配设置与实际安装的模块的类型不相符合时，将无法正常工作。

此外，对于智能功能模块，应使 I/O 点数也相同。

实际安装的模块与 I/O 分配设置的类型不相同时的动作如表 4.1 所示。

表 4.1 实际安装模块与 I/O 分配不相同时的动作一览

实际安装模块	I/O 分配设置	结果
空插槽	智能	作为空插槽处理
所有的模块	空闲	作为空插槽处理

3) 最后的 I/O 地址号

进行了 I/O 分配时，应将最后的 I/O 地址号设置在 I/O 点数的最大值
( 第 2 章) 的范围内。

当设置的最后的 I/O 地址号超出了 I/O 点数的最大值时，将出错 (“MODULE LAYOUT ERROR”)。(在 GX Developer 的系统监视中，I/O 地址显示为 ***)。

(b) CPU 模块自动进行起始 XY 分配时的注意事项

未输入起始 XY 时, CPU 模块将自动进行起始 XY 分配。

因此, 在以下的 1) 或者 2) 的情况下, 各插槽的起始 XY 的设置与 CPU 模块分配的起始 XY 的设置有可能重复。

- 1) 在起始 XY 设置中对前后的 I/O 地址号进行了互换设置
- 2) 设置了起始 XY 的插槽与未设置的 (自动分配的插槽) 混在一起

起始 XY 重复时的示例如图 4.4 所示。

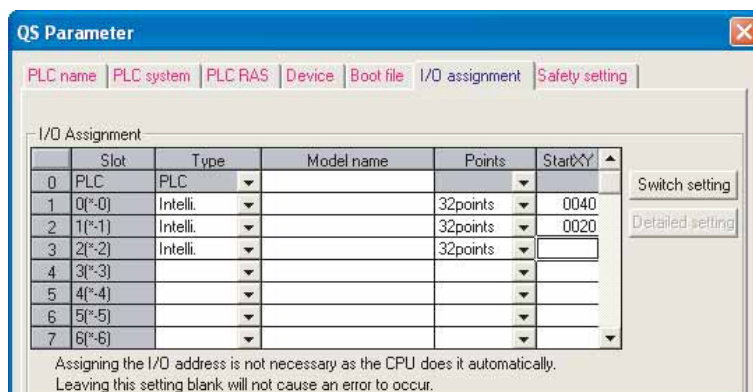


图 4.4 起始 XY 重复的 I/O 分配设置

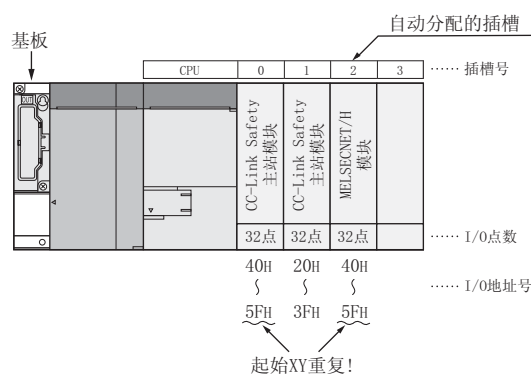


图 4.5 进行了上述 I/O 分配设置时的起始 XY

应注意避免各插槽的起始 XY 重复。

起始 XY 重复时将出错 (“MODULE LAYOUT ERROR”)

4.3.3 I/O 分配示例

通过 GX Developer 进行了 I/O 分配设置时的 I/O 地址号的分配示例如下所示。

(1) 设置安装的模块的 I/O 点数时

将安装了 CC-Link Safety 主站模块、MELSECNET/H 模块的插槽设置为 32 点，当 CC-Link Safety 主站模块、MELSECNET/H 模块由于故障模块而被卸下时，使 I/O 地址不发生变化。

(a) 系统配置及 I/O 地址号的分配

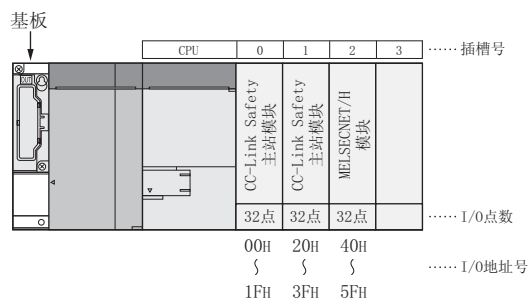


图 4.6 系统配置及 I/O 地址号的分配

(b) 通过 GX Developer 进行 I/O 分配设置

在 GX Developer 的 I/O 分配设置画面中将插槽号 0 至 2 设置为“32 点”。

选择32点。(未选择类型时将成为安装的模块的类型。)

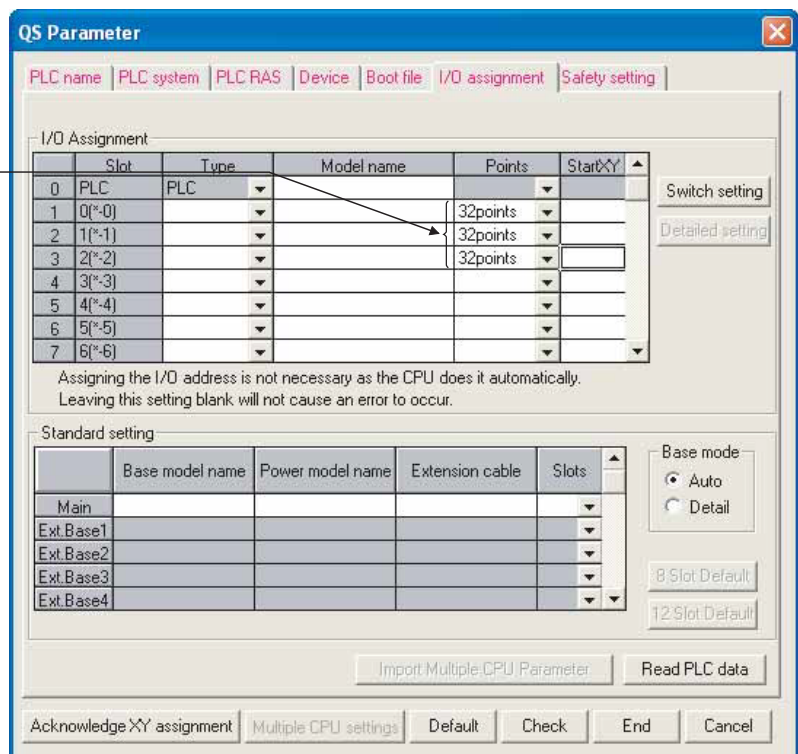


图 4.7 I/O 分配设置

4.4 I/O 地址号的确认

通过 GX Developer 的系统监视，可以确认主基板上安装的模块及 I/O 地址号。
( 6.16 节)

1

概要

2

性能规格

3

顺控程序的执行

4

I/O 地址号的分配

5

关于 CPU 模块中使用的存储器及文件

6

功能

7

与智能功能模块的通信

8

参数

备忘录

[illegible]

第 5 章 关于 CPU 模块中使用的存储器及文件

5.1 CPU 模块中使用的存储器

5.1.1 存储器构成及可存储数据

以下介绍在 CPU 模块中使用的存储器及可存储的数据有关内容。

(1) 存储器构成

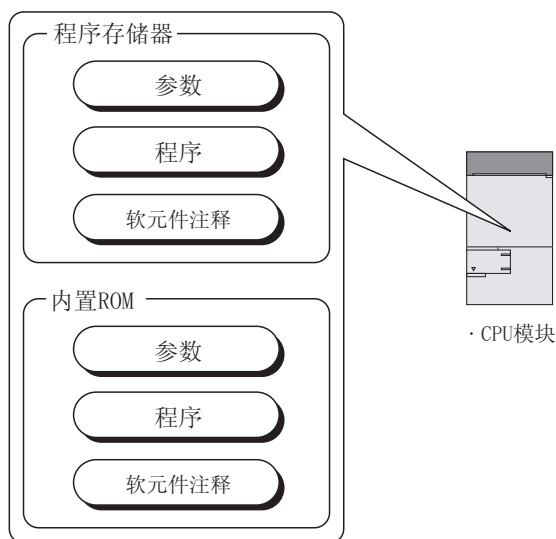


图 5.1 CPU 模块中使用的数据

(a) 程序存储器 (☞ 5.1.2 节)

程序存储器是存储用于 CPU 模块运算的程序的存储器。

(b) 内置 ROM (☞ 5.1.3 节)

内置 ROM 是用于在 CPU 模块中进行引导运行的存储器。

(2) 各存储器中可存储的数据

程序存储器、内置 ROM 中可存储的数据及驱动器号如表 5.1 所示。

表 5.1 可存储的数据及存储目标

驱动器号	CPU 模块内置存储器		文件名及扩展名
	程序存储器	内置 ROM	
	0	4	
参数	◎	○	PARAM.QPA
顺控程序	◎	○	MAIN.QPG
软元件注释	○	○	MAIN.QCD
用户设置的系统区 ^{*1}	○	×	--

◎：必要数据；○：可存储数据；×：不可存储数据

^{*1}：设置在系统中使用的区。(☞ 5.1.2 节 (3)(b))

(3) 存储器容量及是否需要格式化

各存储器的存储容量及是否需要格式化如表 5.2 所示。

表 5.2 是否需要格式化

	QS001CPU	格式化
程序存储器	128k 字节	需要 ^{*1}
内置 ROM	128k 字节	不需要

^{*1}：使用之前必须通过 GX Developer 进行格式化。

5.1.2 关于程序存储器

(1) 关于程序存储器

程序存储器是存储用于 CPU 模块运算的程序的存储器。

内置 ROM 中存储的程序被引导（读取）至程序存储器中进行运算。

(2) 可存储的数据

程序存储器中可以存储参数、程序、软元件注释、用户设置的系统区。

关于程序存储器中可存储的数据的一览表请参阅 5.1.1 节 (2)。

☒ 要 点

程序存储器中存储的数据的总容量超出了程序存储器的程序容量时，应考虑是否减少用户设置的系统区。

(3) 在使用程序存储器之前

使用之前必须通过 GX Developer 对程序存储器进行格式化。

(a) 格式化的执行

执行格式化时，通过 GX Developer 的 [Online(在线)] → [Format PLC memory(可编程控制器存储器格式化)] 显示可编程控制器存储器格式化画面。在可编程控制器存储器格式化画面中，选择“Program memory/Device memory(程序存储器 / 软元件存储器)”。

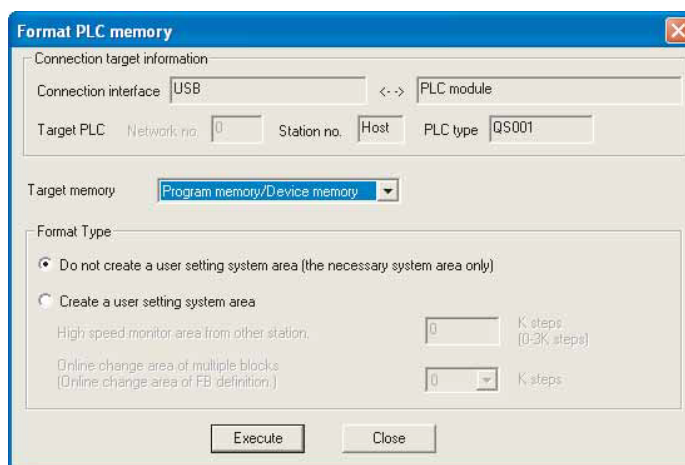


图 5.2 程序存储器的格式化

(b) 用户设置的系统区的创建

进行程序存储器的格式化时，设置用户设置的系统区的容量。

1) 不创建用户设置的系统区

在不创建用户设置的系统区的状况下进行格式化。

2) 创建用户设置的系统区

在格式化时创建用户设置的系统区。

用户设置的系统区中包含有 (表 5.3) 中所示的内容。

表 5.3 用户设置的系统区的类型

系统区类型	内容
用于可以对多个块进行运行中写入的区 (FB 定义的运行中写入用的区)	如果设置了本区域，将可以对多个块进行运行中写入。 关于设置本区域时的可进行运行中写入的块数，请参阅以下手册。  GX Developer 操作手册

☒ 要 点

如果创建用户设置的系统区，可使用的区将会减少相当于创建该区的步数。

通过 GX Developer 的可编程控制器读取画面可以确认存储器容量。

( 本节 (3) (c))

(c) 格式化后的存储器容量的确认

通过 GX Developer 的 [Online(在线)] → [Read from PLC(可编程控制器读取)] 可以确认存储器容量。

- 1) 在可编程控制器读取画面的对象存储器中, 选择 “ Program memory/Device memory(程序存储器 / 软元件存储器) ”。
- 2) 点击 **Free space volume(空余容量)** 按钮。
- 3) 在 Total free space volume(总空余容量栏) 中将显示存储器容量。

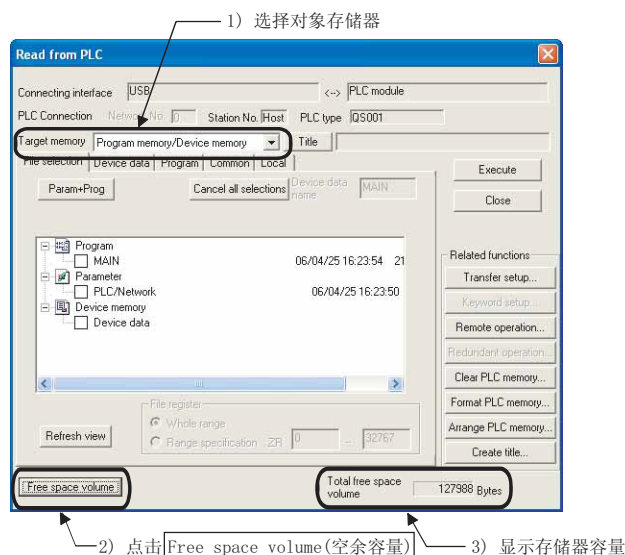


图 5.3 存储器容量的确认步骤

(4) 至程序存储器的写入

对程序存储器进行数据写入时, 通过 GX Developer 的 [Online(在线)] → [Write to PLC(可编程控制器写入)] 显示可编程控制器写入画面。

在可编程控制器写入画面的对象存储器中, 选择 “ Program memory/Device memory(程序存储器 / 软元件存储器) ”, 进行可编程控制器写入。

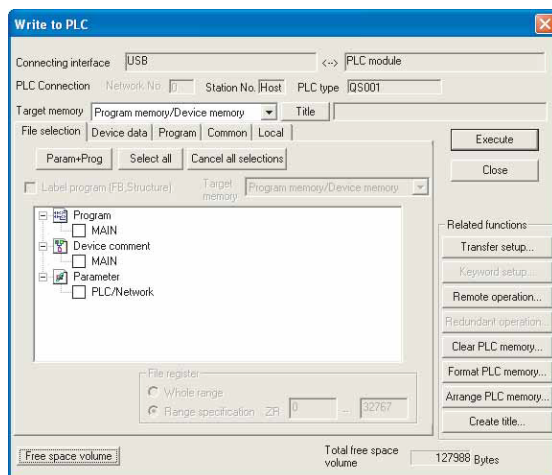


图 5.4 可编程控制器写入画面

☒ 要 点

文件的尺寸有最小单位。(☞ 5.3.4 节)
占用的存储器容量有时会大于实际的文件容量。

5.1.3 关于内置 ROM

(1) 关于内置 ROM

内置 ROM 是用于在 CPU 模块中执行引导运行的存储器。

内置 ROM 可用于在无电池备份的状况下保存程序及参数等。

存储在内置 ROM 中的程序将被引导（读取）到程序存储器（ 5.1.2 节）中进行运算。

(2) 可存储的数据

内置 ROM 中可以存储参数、程序、软元件注释。

关于各存储器中可存储的数据的一览表，请参阅 5.1.1 节 (2)。

(3) 存储器容量的确认

通过 GX Developer 的 [Online(在线)] → [Read from PLC(可编程控制器读取)] 可以确认存储器容量。

- 1) 在可编程控制器写入画面的对象存储器中，选择 “Standard ROM(内置 ROM)”。
- 2) 点击  按钮。
- 3) 在 Total free space volume(总空余容量栏) 中将显示存储器容量。

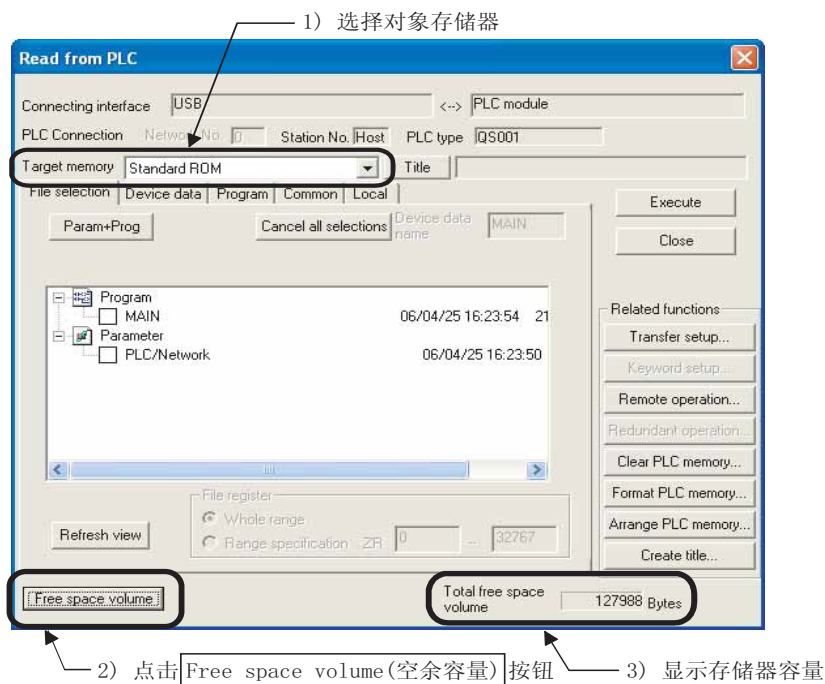


图 5.5 存储器容量的确认步骤

(4) 至内置 ROM 的写入

关于至内置 ROM 的写入，请参阅 5.1.4 节 (3)。

☒ 要 点

文件的尺寸有最小单位。(☞ 5.3.4 节)

占用的存储器容量有时会大于实际的文件容量。

(5) 内置 ROM 中存储的程序的使用方法

由于在内置 ROM 中存储的程序中不能执行运算，因此内置 ROM 中存储的程序被引导 (读取) 到程序存储器中使用。(☞ 5.1.4 节)

5.1.4 内置 ROM 的程序的执行（引导运行）及写入

(1) 内置 ROM 的程序的执行（引导运行）

(a) 关于内置 ROM 的程序的执行

CPU 模块进行程序存储器中存储的程序的运算。

在内置 ROM 中存储的程序中不能进行运算。

内置 ROM 中存储的程序被引导（读取）到程序存储器中进行运算。

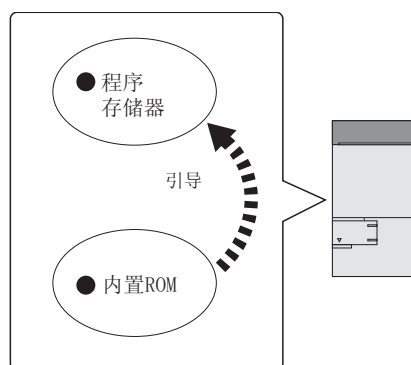


图 5.6 引导运行

1) 安全模式的情况下

在安全模式下，与通过 GX Developer 进行的引导设置无关，进行引导运行。

2) 测试模式的情况下

在 GX Developer 的引导设置中设置为“执行引导运行”，通过写入到内置 ROM 可以进行引导运行。

☒ 要 点

测试模式时，对程序存储器的参数、程序进行了调试时，从测试模式切换至安全模式时应进行至内置 ROM 的写入。

(2) 执行引导运行前的步骤及引导运行的终止（测试模式时）

(a) 执行引导运行前的步骤

执行引导运行前的步骤如下所示。

- 1) 通过 GX Developer 创建程序
创建执行引导运行的程序。
- 2) 通过 GX Developer 进行引导设置
在可编程控制器参数的引导文件设置中，设置为“Do boot from Standard ROM(执行来自于内置 ROM 的引导)”。

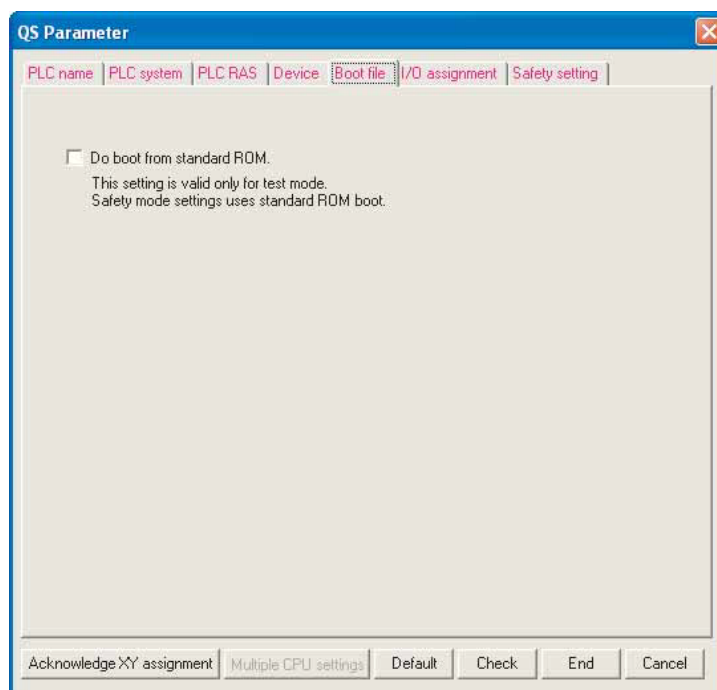


图 5.7 引导文件设置

1

概要

2

性能规格

3

顺控程序的执行

4

I/O 地址号的分配

5

关于 CPU 模块中使用的存储器及文件

6

功能

7

与智能功能模块的通信

8

参数

- 3) 通过 GX Developer 对内置 ROM 进行写入
 - 通过GX Developer的[Online(在线)]→[Write to PLC(可编程控制器写入)]将文件写入到程序存储器中。
 - 将写入到存储器中的文件通过GX Developer的[Online(在线)]→[Write to PLC (Flash ROM) (可编程控制器写入 (快闪卡))] → [Write the program memory to ROM...(程序存储器的 ROM 化)] 写入到内置 ROM 中。
( 本节 (3))

- 4) 程序的执行

执行了以下操作时，将执行来自于内置 ROM 的引导。

 - 可编程控制器的电源的重启
 - 通过 CPU 模块的 RUN/STOP/RESET 开关进行的复位解除
 - 通过 GX Developer 进行的远程操作

- 5) 引导是否正常结束的确认

通过特殊继电器 (SM660) 的状态可以确认引导是否正常结束。
关于特殊继电器，请参阅附录 1。

(b) 终止引导运行时的操作

终止引导运行，通过写入到程序存储器中的参数程序执行运行时，应通过 GX Developer 进行以下操作。

- 1) 在可编程控制器参数的引导文件设置中，取消 “ Do boot from Standard ROM(执行来自于内置 ROM 的引导) ” 的选中。
- 2) 对程序存储器进行参数、顺控程序的写入。
- 3) 执行 [Online(在线)] → [Write to PLC (Flash ROM) (可编程控制器写入 (快闪卡))] → [Write the program memory to ROM...(程序存储器的 ROM 化)]。
(内置 ROM 的参数、顺控程序将被删除。)

(c) 执行内置 ROM 的程序时的注意事项

- 1) 关于内置 ROM 中存储的文件

进行引导运行时，应将以下文件存储到内置 ROM 中。

 - 参数 *1
 - 程序 *1
 - 软元件注释

*1 : 必须存储到内置 ROM 中。

- 2) 关于引导运行时的运行中写入

进行来自于内置 ROM 的引导运行时，即使执行程序存储器的程序的运行中写入，更改内容也不会被存储到引导源的内置 ROM 的程序中。
因此，应在变为 STOP 状态时进行至内置 ROM 的写入 ( 本节 (3))。
- 3) 进行电源 OFF → ON 或者复位解除时程序存储器的内容发生了变化的情况

将顺控程序写入到程序存储器中后进行了可编程控制器的电源 OFF → ON 或 CPU 模块的复位解除时，程序存储器的内容被改写的情况下，可能是由于正在进行引导运行所致。
应参照本节 “ (2) (b) 终止引导运行时的操作 ” 终止引导运行。

(3) 至内置 ROM 的写入

至内置 ROM 的写入是通过将程序存储器的文件批量地复制到内置 ROM 中进行的。

(a) 写入前

在将文件写入到内置 ROM 之前，应确认以下几点。

1) 内置 ROM 内的文件保存

将文件写入到内置 ROM 中时，内置 ROM 中存储的所有文件将被自动删除。

在写入到内置 ROM 之前，应使用 GX Developer 的 [Online(在线)] → [Read from PLC(可编程控制器读取)]，预先将存储的文件通过 GX Developer 进行保存。

2) 写入文件的准备

将文件写入到内置 ROM 中时，由于内置 ROM 中存储的所有文件将被自动删除，因此应将存储的所有文件预先备份。

(b) 写入步骤

将文件写入到内置 ROM 中的步骤如下所示。

1) 选择 GX Developer 的 [Online(在线)] → [Write to PLC (Flash ROM) (可编程控制器写入 (快闪卡)] → [Write the program memory to ROM...(程序存储器的 ROM 化)]。

2) 显示程序存储器的 ROM 化画面。

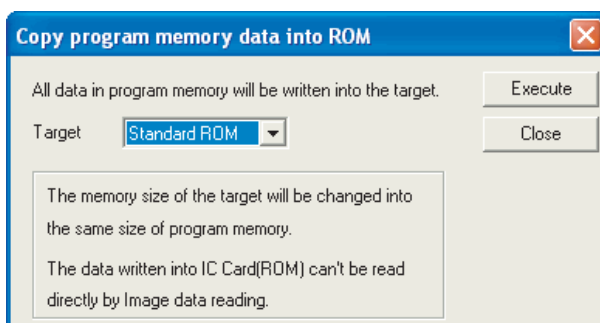


图 5.8 程序存储器的 ROM 化画面

3) 选择写入目标，将程序存储器的文件复制到内置 ROM 中。

(4) 对内置 ROM 的文件进行添加 / 更改 (测试模式时)

将文件写入到内置 ROM 中时，内置 ROM 中存储的所有文件将被自动删除，因此不能对存储的文件直接进行添加、更改。

进行添加、更改时应通过以下方法进行。

- 1) 通过 GX Developer 的 [Online(在线)] → [Read from PLC(可编程控制器读取)] 读取内置 ROM 的所有文件。
- 2) 对读取的文件进行添加、更改。
- 3) 将进行了添加、更改的文件写入到程序存储器中。
- 4) 通过 [Online(在线)] → [Write to PLC (Flash ROM) (可编程控制器写入 (闪存卡)] → [Write the program memory to ROM...(程序存储器的 ROM 化)] 将文件复制到内置 ROM 中。

(5) 注意事项 (测试模式时)

(a) 关于 GX Developer 的通信时间确认时间的设置

将文件写入到内置 ROM 中时，GX Developer 的通信时间确认时间为 180 秒以下时，以 180 秒进行确认。

5.2 程序文件的构成

程序文件是由文件标题、执行程序、运行中写入预留容量（步）所构成。

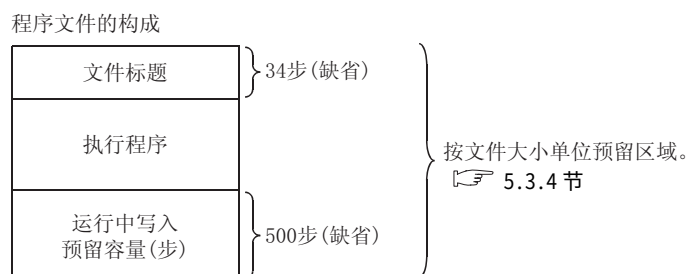


图 5.9 程序文件的构成

(1) 各构成的详细内容

CPU 模块的程序存储器中存储的程序容量为上述 3 种类型区域的合计。

(a) 文件标题

是存储文件名、文件大小、文件创建日期等的区域。

文件标题的大小根据可编程控制器参数的软元件设置可在 26 ~ 35 步 (104 ~ 140 字节) 的范围内更改。(缺省为 34 步)

(b) 执行程序

是存储创建的程序的区域。

(c) 运行中写入预留容量（步）

是通过 GX Developer 进行增加步数的运行中写入时使用的区域。

通过 GX Developer 进行增加步数的运行中写入时，将显示运行中写入预留容量（步）的剩余步数。

1) 缺省的运行中写入预留容量（步）

缺省设置为 500 步 (2000 字节)。

2) 运行中写入预留容量（步）的更改

运行中写入预留容量（步）可以通过 GX Developer([Online(在线)] → [Write to PLC(可编程控制器写入)] 的 <<Program>> 选项卡) 进行更改。

此外，在进行运行中写入时，在运行中写入预留容量（步）不足的情况下，可以重新设置运行中写入预留容量（步）。(6.14.1 节)

(2) 通过 GX Developer 显示程序容量

通过 GX Developer 进行编程时，如图 5.10 所示以步数显示程序容量（文件标题容量及创建的程序的步数的合计）。

创建程序时，可以确认创建的程序的容量。

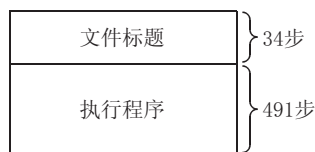


图 5.10 程序容量的显示

☒ 要 点

1. 通过 GX Developer 进行编程时显示的程序容量为文件标题 + 执行程序的容量，不包含运行中写入预留容量（500 步）。

（例）执行程序部分为 491 步的程序在 GX Developer 上的容量如下图所示。
（文件标题的缺省为 34 步）



GX Developer 上的显示：
34 步 + 491 步 = 525 步

图 5.11 GX Developer 上的文件状态

2. 在程序存储器上由于文件是以文件大小单位进行存储的，因此通过 GX Developer 编程时显示的程序容量与 CPU 模块上程序存储器的容量有可能会不一样。（☞ 5.3.4 节）

5.3 通过 GX Developer 进行文件操作及处理时的注意事项

5.3.1 文件操作

通过 GX Developer 的在线操作可以对存储在程序存储器、内置 ROM 中的文件进行文件操作。

但是，根据安全 CPU 动作模式、CPU 模块的 RUN/STOP 状态的不同，可执行的文件操作也有所不同。

(☞ 6.2.5 节)

5.3.2 文件处理时的注意事项

(1) 关于文件操作时的电源 OFF (包括复位)

通过 GX Developer 进行文件操作时进行了可编程控制器电源的 OFF 或者 CPU 模块的复位时，各存储器的文件将成为不定值。

在通过 GX Developer 进行文件操作时不要进行可编程控制器电源的 OFF 或者 CPU 模块的复位。

1

概要

2

性能规格

3

顺控程序的执行

4

I/O 地址号的分配

5

关于 CPU 模块中使用的存储器及文件

6

功能

7

与智能功能模块的通信

8

参数

5.3.3 文件的容量

使用程序存储器、内置 ROM 时，应通过表 5.4 大致计算出各文件的大小。

表 5.4 文件的存储容量的计算

功能	大致文件容量 (单位: 字节)
驱动器目录	70
参数	缺省: 316 (根据参数的设置而增加) 参考 引导设置 → 100 有 MELSECNET/H 设置 → 最大增加 156 有 CC-Link 设置 → $22 + 572 \times (\text{安全主站模块个数}) + 76 \times (\text{安全远程站个数}) + 4 \times (\text{安全远程站参数设置数})$
顺控程序	$134^* + (4 \times ((\text{步数}) + (\text{运行中写入预留容量})))$
软元件注释	$80 + (\text{各软元件注释数据容量的合计})$ • $1 \text{ 各软元件注释数据容量} = 10 + 10210 \times a + 40 \times b$ • $a : ((\text{软元件点数})/256) \text{ 的商}$ • $b : ((\text{软元件点数})/256) \text{ 的余数}$
多个块运行中写入设置	格式化时的设置值 (0/1.25k/2.5k)

* : 缺省值为 134 (根据参数的设置而增加。)

5.3.4 文件的大小单位

(1) 关于文件的大小单位

将文件写入到存储区时的最小单位称为文件大小单位。
CPU 模块的文件大小单位为 4 字节。

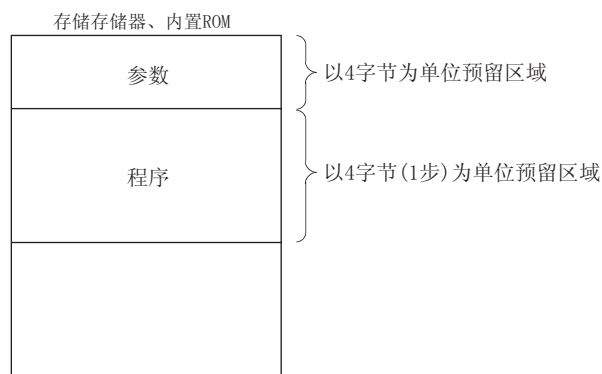


图 5.12 程序存储器、内置 ROM 的文件大小单位

第 6 章 功能

本章介绍 CPU 模块的功能有关内容。

6.1 功能一览

CPU 模块的功能一览如表 6.1 所示。

表 6.1 CPU 模块的功能一览

项目	内容	安全 CPU 动作模式		参阅章节
		安全模式	测试模式	
安全 CPU 动作模式	是选择将 CPU 模块作为安全装置的一部分进行正常运行，还是通过程序更改及软件测试功能进行 CPU 模块的维护的功能。	○	○	6.2 节
CPU 存取密码	是防止通过误连接的 GX Developer 进行的非法操作的功能。	○	○	6.3 节
可编程控制器存储器初始化	是删除写入到 CPU 模块中的用户数据的功能。执行可编程控制器存储器初始化时，将变为产品出厂时的状态。	○	○	6.4 节
测试模式连续运行防止设置	是防止可编程控制器系统在测试模式下长时间连续运行的功能。	×	○	6.5 节
ROM 写入次数的确认	是确认至 ROM 的写入次数的功能。	○	○	6.6 节
自诊断功能	是进行 CPU 模块自身有无异常的诊断的功能。	○	○	6.7 节
操作 / 故障历史记录	是记录从外部对 CPU 模块进行的操作及 CPU 模块以前发生的自诊断出错的功能。	○	○	6.8 节
恒定扫描	使程序按一定周期执行的功能。	○	○	6.9 节
进行了 STOP ↔ RUN 时的输出状态的选择功能	是选择将 CPU 模块从 STOP 状态变为 RUN 状态时的输出 (Y) 的状态 (STOP 前的输出的再输出 / 执行运算后的输出) 的功能。	○	○	6.10 节
时钟功能	是执行 CPU 模块内置时钟的功能。	○	○	6.11 节
远程 RUN/STOP	是停止或执行 CPU 模块的运算的功能。	○	○	6.12.1 节
远程 RESET	是 CPU 模块处于 STOP 状态时，复位 CPU 模块的功能。	○	○	6.12.2 节
监视功能	通过 GX Developer 读取 CPU 模块的程序、软元件的状态的功能。	○	○	6.13 节
运行中写入	是将程序写入到正在运行的 CPU 模块中的功能。	×	○	6.14 节
看门狗定时器	是监视由于 CPU 模块的硬件、程序异常等导致运算迟滞的功能。	○	○	6.15 节
系统显示	是连接 GX Developer 进行系统配置监视的功能。	○	○	6.16 节
LED 显示	是通过 CPU 模块前面的 LED 显示 CPU 模块的动作状态的功能。	○	○	6.17 节

○：可以使用 ×：不可使用

6.2 安全 CPU 动作模式

6.2.1 安全 CPU 动作模式

安全 CPU 动作模式中包含有“安全模式”及“测试模式”这 2 种模式。
安全 CPU 动作模式的切换是通过 GX Developer 操作进行的。

(1) 安全模式

是安全系统的正常运行时使用的模式。

在安全模式下，为了保护正常运行中的系统，将禁止进行可编程控制器写入、软元件测试等使安全可编程控制器的控制发生变化的操作。

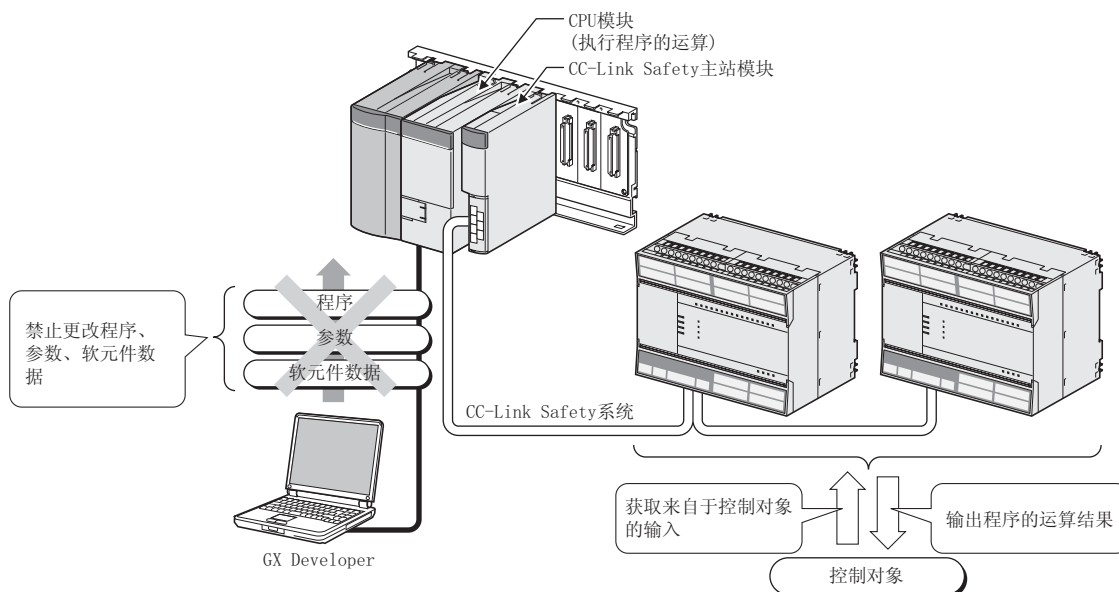


图 6.1 安全模式的动作

(2) 测试模式

是对系统进行启动及维护时使用的模式。

可以使用可编程控制器写入及软元件测试等 GX Developer 的所有功能。

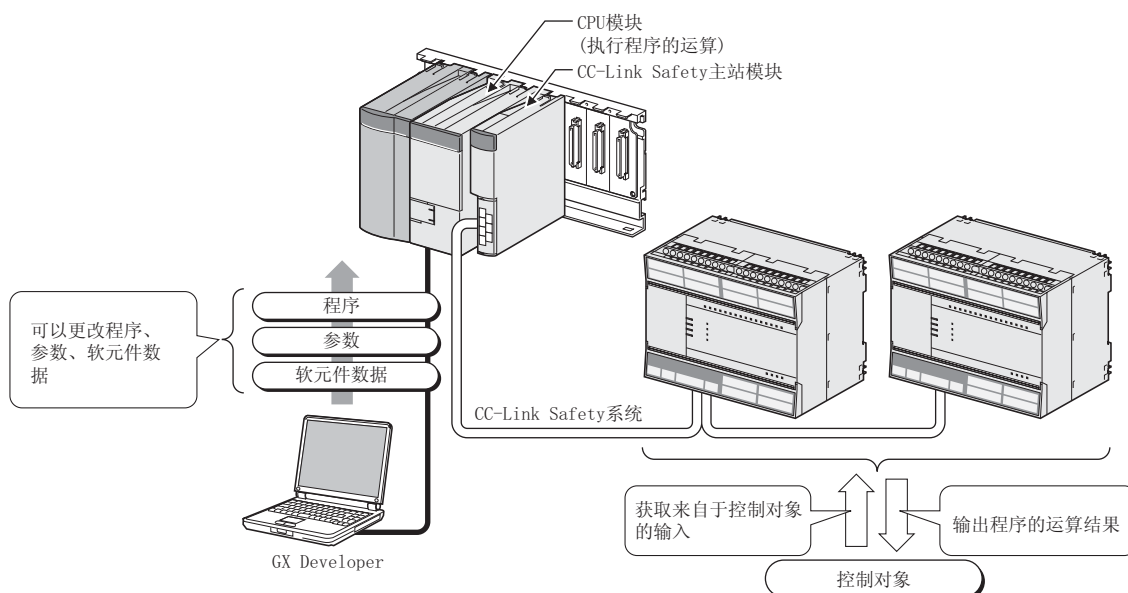


图 6.2 测试模式的动作

备注

关于安全模式及测试模式下可执行的 GX Developer 的操作的有关内容，请参阅 GX Developer 的操作手册（安全可编程控制器篇）。

(3) 安全 CPU 动作模式的切换

安全 CPU 动作模式切换时的状态如图 6.3 所示。

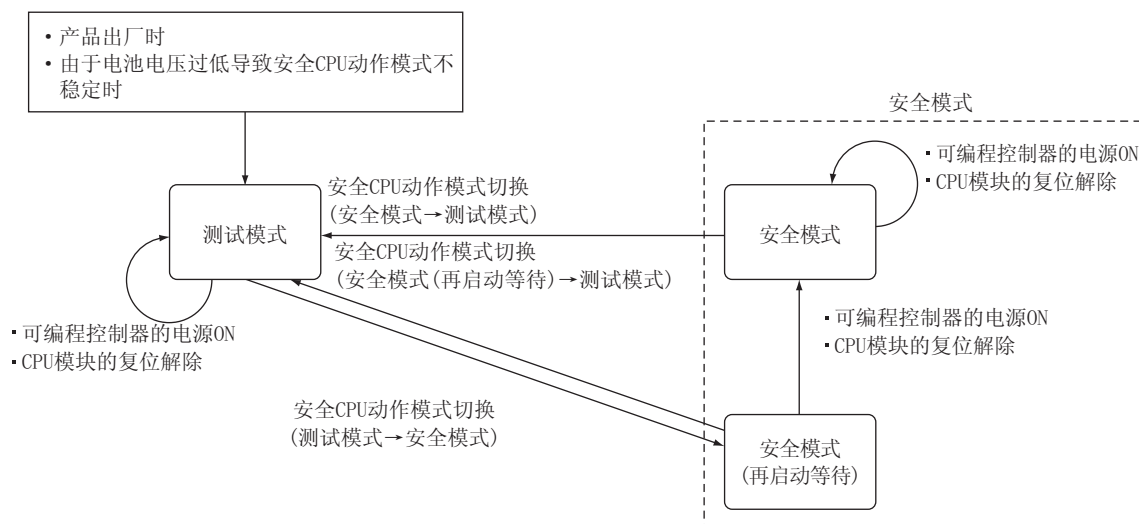


图 6.3 安全 CPU 动作模式切换时的状态

☒ 要 点

1. 安全 CPU 动作模式的信息通过 CPU 模块的电池保存。
使用 CPU 模块时，应连接 CPU 模块附带的电池。
2. 在“安全模式（再启动等待）”中，不能执行程序。
(即使将 RUN/STOP/RESET 开关由 STOP 变为 RUN，或者通过 GX Developer 进行远程 RUN，CPU 模块也不能变为 RUN 状态。)
3. 在以下情况下在测试模式下启动。
 - 购买产品后第 1 次接通电源时
 - 由于电池电压过低导致安全 CPU 动作不稳定时 (操作 / 故障历史记录中将存储操作内容“OP001:SYSTEM INITIALIZE OPERATION MODE”。)

6.2.2 安全 CPU 动作模式的确认

通过以下方法可以确认 CPU 模块的安全 CPU 动作模式。

- 通过 CPU 模块前面的 LED 进行确认
- 通过 GX Developer 的在线操作画面进行确认
- 通过特殊继电器、特殊寄存器进行确认

(1) 通过 CPU 模块前面的 LED 进行确认

通过 CPU 模块前面的“ALIVE”LED 及“TEST”LED，可以确认当前的安全 CPU 动作模式。

表 6.2 通过“ALIVE”LED 及“TEST”LED 确认安全 CPU 的动作模式

测试模式	安全模式（再启动等待）	安全模式
亮灯 → ALIVE  TEST  RUN ☐ USER ☐ ERR. ☐ BAT. ☐	亮灯 → ALIVE  TEST  RUN ☐ USER ☐ ERR. ☐ BAT. ☐	亮灯 → ALIVE  TEST  RUN ☐ USER ☐ ERR. ☐ BAT. ☐

(2) 通过 GX Developer 的在线操作画面进行确认

在 GX Developer 的在线操作画面（可编程控制器诊断、远程操作等）中，显示 CPU 模块的当前的安全 CPU 动作模式。

通过 GX Developer 进行远程操作等时，可以确认安全 CPU 动作模式。

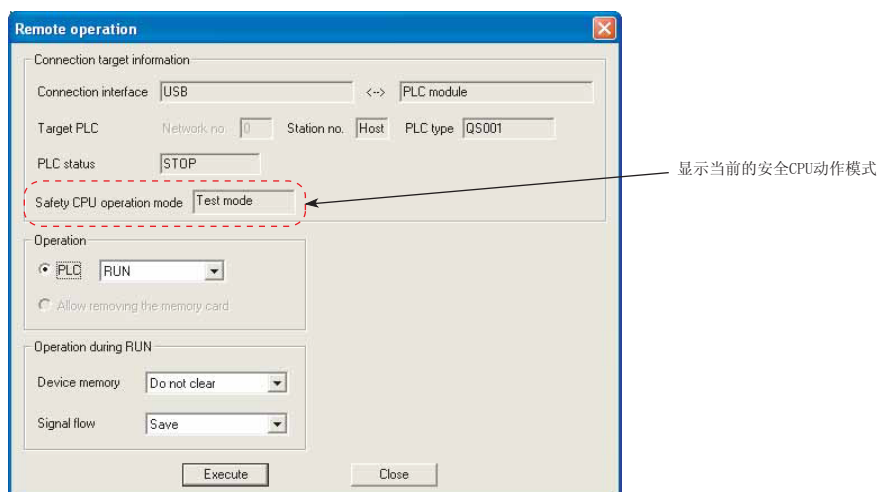


图 6.4 通过 GX Developer 显示安全 CPU 动作模式

(3) 通过特殊继电器、特殊寄存器进行确认

当前的安全 CPU 动作模式存储在 CPU 模块的特殊继电器的 SM560 (测试模式标志) 及特殊寄存器的 SD560 (安全 CPU 动作模式) 中。

通过程序使用 SM560、SD560，可以在外部显示安全 CPU 动作模式。

此外，在 GX Developer 中监视 SM560、SD560，也可以确认安全 CPU 动作模式。

表 6.3 用于确认安全 CPU 动作模式的特殊继电器、特殊寄存器

软元件名	名称	说明
SM560	测试模式标志	显示当前的安全 CPU 模式是否为测试模式。 • OFF: 安全模式或者安全模式 (再启动等待) • ON: 测试模式
SD560	安全 CPU 动作模式	显示当前的安全 CPU 动作模式。 • 0: 安全模式 • 1: 测试模式 • 2: 安全模式 (再启动等待)

6.2.3 安全 CPU 动作模式的切换

安全 CPU 动作模式的切换是通过 GX Developer 的“安全 CPU 动作模式切换”操作进行的。

(1) 安全 CPU 动作模式的切换条件

在如表 6.4 所示的状态时可以执行安全 CPU 动作模式的切换。

表 6.4 安全 CPU 动作模式切换的执行条件

安全 CPU 动作模式的切换条件	测试模式→安全模式	安全模式→测试模式
CPU 动作状态	处于 STOP 状态 (不包括由于停止出错导致的 STOP 状态)	处于 STOP 状态 (包括由于停止出错导致的 STOP 状态)
GX Developer 与程序存储器的程序、参数	一致	-
通过其它的 GX Developer 进行的可编程控制器写入及软元件测试等的操作	未执行	-
通过其它的 GX Developer 进行的安全 CPU 动作模式切换操作	未执行	未执行

(2) 安全 CPU 动作模式切换步骤

通过 GX Developer 的“安全 CPU 动作模式切换”操作进行安全 CPU 动作模式切换的步骤如下所示。

(a) 测试模式至安全模式的切换

通过 GX Developer 进行测试模式至安全模式的切换步骤如图 6.5 所示。

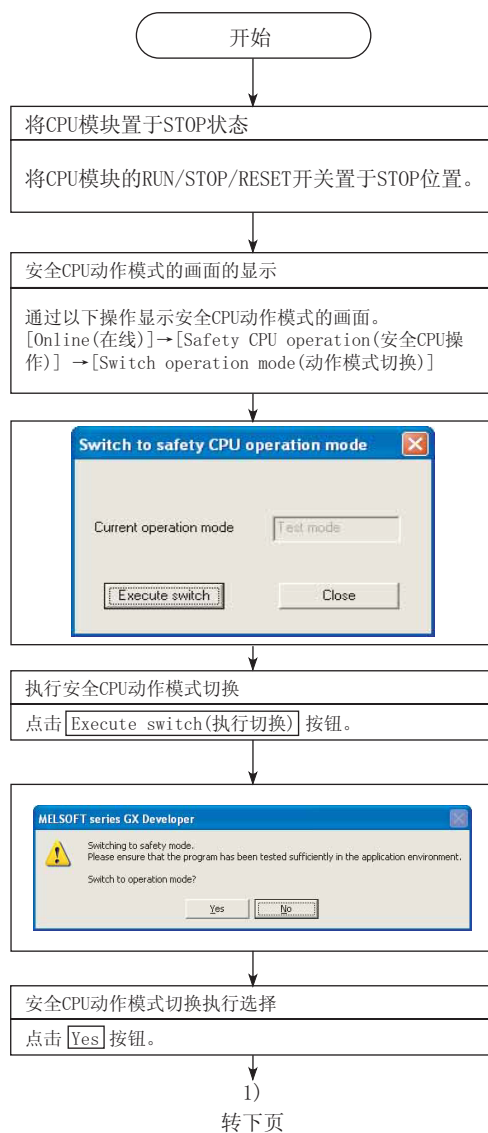


图 6.5 测试模式至安全模式的切换

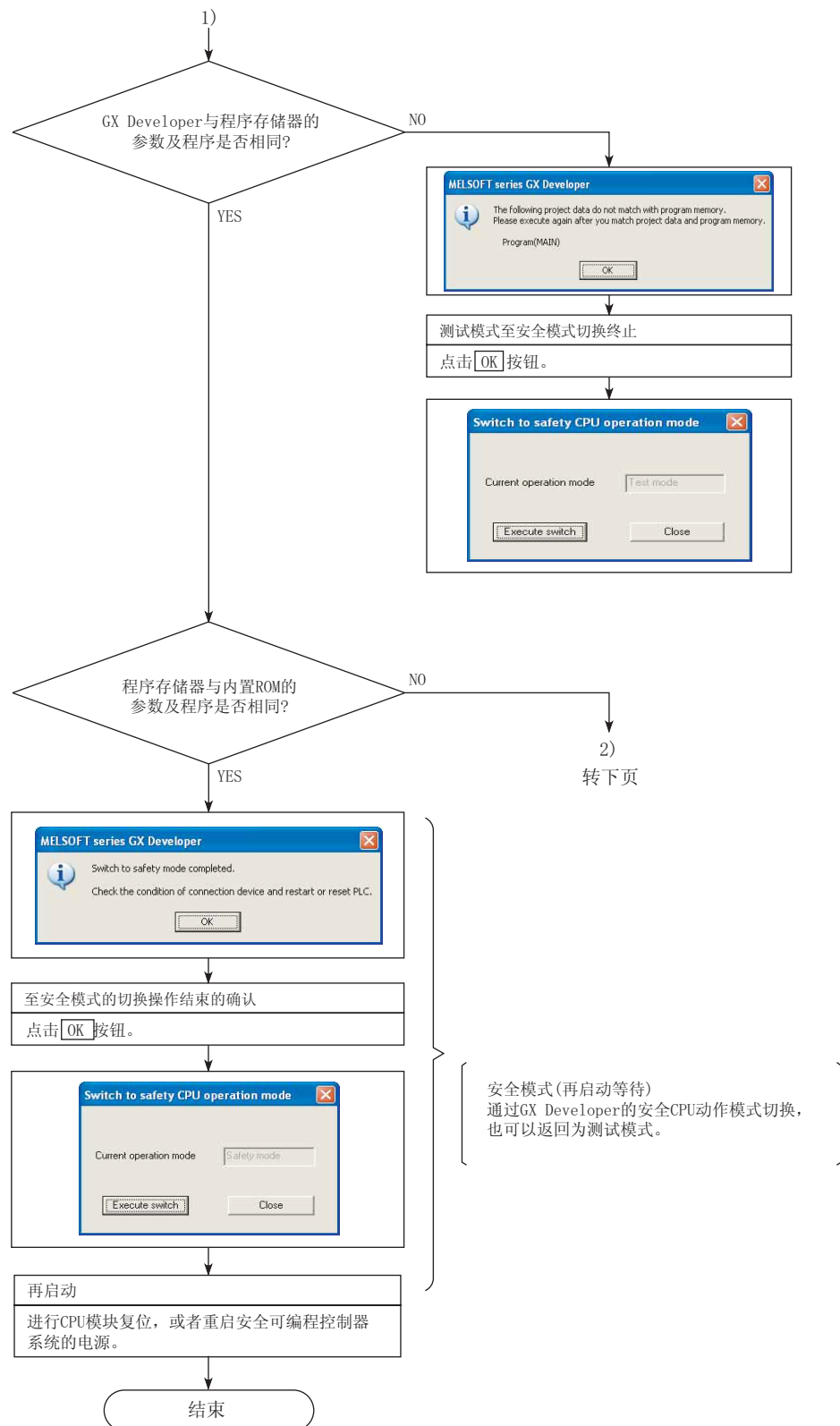


图 6.5 测试模式至安全模式的切换 (续)

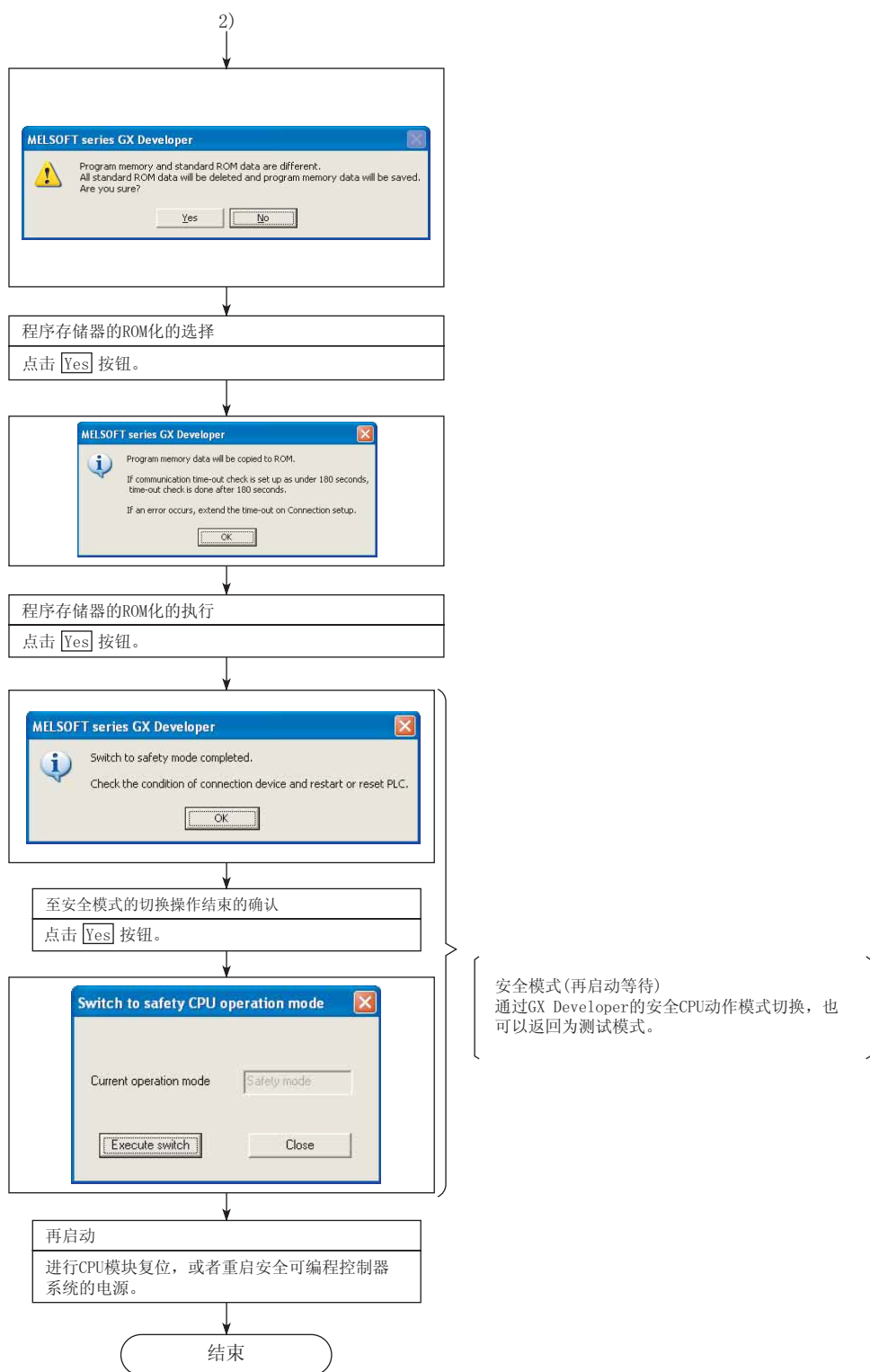


图 6.5 测试模式至安全模式的切换 (续)

(b) 安全模式至测试模式的切换

通过 GX Developer 进行安全模式至测试模式的切换步骤如图 6.6 所示。

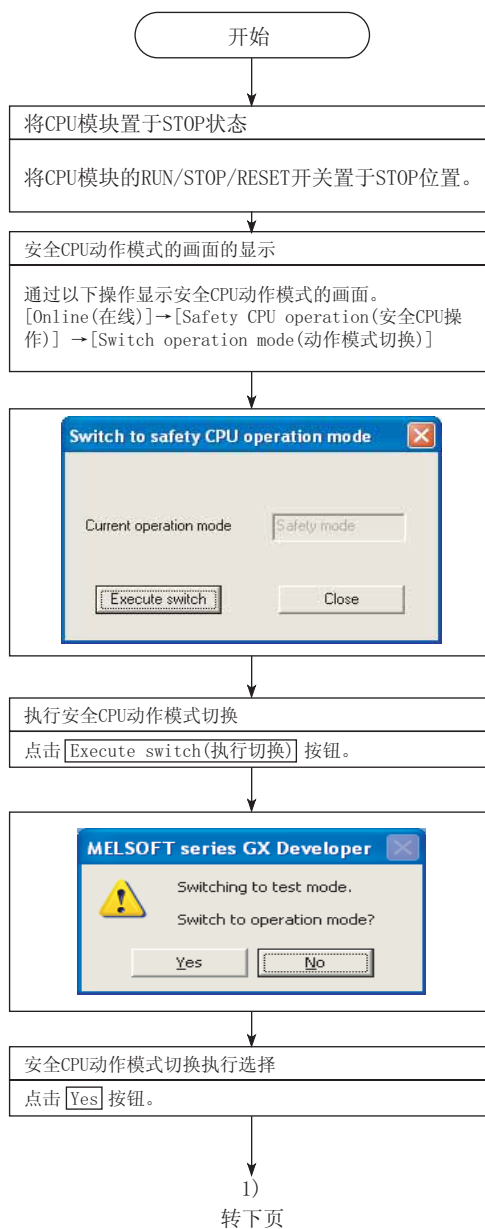


图 6.6 安全模式至测试模式的切换



图 6.6 安全模式至测试模式的切换（续）

1

概要

2

性能规格

3

顺控程序的执行

4

I/O 地址号的分配

5

关于 CPU 模块中使用的存储器及文件

6

功能

7

与智能功能模块的通信

8

参数

6.2.4 安全 CPU 动作模式、CPU 动作状态的各功能的动作

安全 CPU 动作模式、CPU 动作状态的各功能的执行可否如表 6.5 所示。

表 6.5 安全 CPU 动作模式、CPU 动作状态的各功能的执行可否

No.	安全 CPU 动作模式			测试模式				
	CPU 动作状态			RUN 状态	STOP 状态	停止出错 *1	STOP → RUN 切换中	初始化 处理中
1	执行程序			○	×	×	×	×
2	CC-Link Safety	CPU → CC-Link 刷新	RY, RWw	○	○ *2	×	×	×
			SB, SW	○	○	×	×	×
		CC-Link → CPU 刷新	RX, RWr	○	○	×	×	×
			SB, SW	○	○	○	×	×
		CC-Link 远程 I/O 站动作	RY → 外部输出	○	○	○ (OFF 输出)	○	×
			外部输入 → RX	○	○	○	○	×
3	MELSECNET/H	CPU → MELSECNET/H 刷新	B, W	○	○	×	×	×
			SB, SW	○	○	×	×	×
		MELSECNET/H → CPU 刷新	B, W	○	○	×	×	×
			SB, SW	○	○	○	×	×

○：功能动作，×：功能不动作，—：组合不存在

*1：表示由于中度异常或者重度异常导致发生停止出错。

关于中度异常、重度异常的有关内容请参阅以下手册。

 QSCPU 用户手册（硬件设计 / 维护点检篇）

*2：根据 CC-Link Safety 参数的“CPU STOP 时设置”，变为以下动作。

- CPU STOP 时设置为“强制清除”时：输出 OFF
- CPU STOP 时设置为“不强制清除”时：输出 RY 的状态

	安全模式（再启动等待）					安全模式				
	RUN 状态	STOP 状态	停止出错 *1	STOP → RUN 切换中	初始化 处理中	RUN 状态	STOP 状态	停止出错 *1	STOP → RUN 切换中	初始化 处理中
	—	×	×	—	—	○	×	×	×	×
	—	○ (输出 OFF)	×	—	—	○	○ (输出 OFF)	×	×	×
	—	○	×	—	—	○	○	×	×	×
	—	○	×	—	—	○	○	×	×	×
	—	○	○	—	—	○	○	○	×	×
	—	○	○ (输出 OFF)	—	—	○	○	○ (输出 OFF)	○	×
	—	○	○	—	—	○	○	○	○	×
	—	○	×	—	—	○	○	×	×	×
	—	○	×	—	—	○	○	×	×	×
	—	○	×	—	—	○	○	×	×	×
	—	○	○	—	—	○	○	○	×	×

○：功能动作，×：功能不动作，—：组合不存在

*1：表示由于中度异常或者重度异常导致发生停止出错。
关于中度异常、重度异常的有关内容请参阅以下手册。

☞ QSCPU 用户手册（硬件设计 / 维护点检篇）

6.2.5 可以通过 GX Developer 对 CPU 模块进行的在线操作

可以通过 GX Developer 对 CPU 模块进行的在线操作如表 6.6 所示。

表 6.6 可以通过 GX Developer 对 CPU 模块进行的在线操作

No.	安全 CPU 动作模式		测试模式					
	CPU 动作状态		RUN 状态	状态 STOP 状态	停止出错 *1	STOP → RUN 切换中	初始化处理中	
1	文件操作	可编程控制器写入	×	○	○	×	×	
		可编程控制器读取	○	○	○	×	×	
		可编程控制器校验	○	○	○	×	×	
		可编程控制器数据删除	×	○	○	×	×	
2	驱动器操作	可编程控制器存储器整理	×	○	○	×	×	
		可编程控制器存储器格式化	×	○	○	×	×	
		驱动器目录登录	×	○	○	×	×	
		驱动器目录删除	×	○	○	×	×	
		程序存储器的 ROM 化	×	○	○	×	×	
3	可编程控制器存储器操作	可编程控制器存储器清除	×	○	○	×	×	
4	程序更改	程序的运行中写入	○	○	○	×	×	
		TC 设置值的运行中写入	○	○	○	×	×	
5	监视	梯形图监视	○	○	○	×	×	
		软元件批量监视	○	○	○	×	×	
		软元件登录监视	○	○	○	×	×	
		缓冲存储器批量监视	○	○	○	×	×	
		程序一览监视	○	○	○	×	×	
6	软元件测试		○	○	○	×	×	
7	远程操作	远程 RUN	○	○	×	×	×	
		远程 STOP	○	○	×	×	×	
		远程复位	×	○	○	×	×	
8	时钟设置	时钟数据的读取	○	○	○	×	×	
		时钟数据的写入	○	○	○	×	×	
9	诊断	可编程控制器诊断	○	○	○	×	×	
		可编程控制器诊断历史记录清除	○	○	○	×	×	
		MELSECNET(II)/10/H 诊断	○	○	○	×	×	
		CC-Link/CC-Link/LT 诊断	○	○	○	×	×	
		系统监视	○	○	○	×	×	
10	安全 CPU 动作模式变更	至安全模式的变更	×	○	×	×	×	
		至测试模式的变更	—	—	—	—	—	
11	CPU 存取密码	CPU 存取密码登录	×	○	○	×	×	
		CPU 存取密码更改	×	○	○	×	×	
12	安全 CPU 操作	可编程控制器存储器初始化	×	○	○	×	×	

○：功能动作，×：功能不动作，—：组合不存在

*1：表示由于中度异常或者重度异常导致发生停止出错。

关于中度异常、重度异常的有关内容请参阅以下手册。

 QSCPU 用户手册（硬件设计 / 维护点检篇）

	安全模式 (再启动等待)					安全模式				
	RUN 状态	STOP 状态	停止出错 *1	STOP → RUN 切换中	初始化处理中	RUN 状态	STOP 状态	停止出错 *1	STOP → RUN 切换中	初始化处理中
	—	×	×	—	—	×	×	×	×	×
	—	○	○	—	—	○	○	○	×	×
	—	○	○	—	—	○	○	○	×	×
	—	×	×	—	—	×	×	×	×	×
	—	×	×	—	—	×	×	×	×	×
	—	×	×	—	—	×	×	×	×	×
	—	×	×	—	—	×	×	×	×	×
	—	×	×	—	—	×	×	×	×	×
	—	×	×	—	—	×	×	×	×	×
	—	×	×	—	—	×	×	×	×	×
	—	×	×	—	—	×	×	×	×	×
	—	×	×	—	—	×	×	×	×	×
	—	×	×	—	—	×	×	×	×	×
	—	×	×	—	—	×	×	×	×	×
	—	○	○	—	—	○	○	○	×	×
	—	○	○	—	—	○	○	○	×	×
	—	○	○	—	—	○	○	○	×	×
	—	○	○	—	—	○	○	○	×	×
	—	○	○	—	—	○	○	○	×	×
	—	×	×	—	—	×	×	×	×	×
	—	×	×	—	—	○	○	×	×	×
	—	×	×	—	—	○	○	×	×	×
	—	○	○	—	—	×	○	○	×	×
	—	○	○	—	—	○	○	○	×	×
	—	×	×	—	—	○	○	○	×	×
	—	×	×	—	—	○	○	○	×	×
	—	—	—	—	—	—	—	—	—	×
	—	○	○	—	—	×	○	○	×	×
	—	×	×	—	—	×	×	×	×	×
	—	×	×	—	—	×	×	×	×	×
	×	○	○	×	×	×	○	○	×	×

○: 功能动作 . ×: 功能不动作 . -: 组合不存在

*1: 表示由于中度异常或者重度异常导致发生停止出错。
关于中度异常、重度异常的有关内容请参阅以下手册。

 QSCPU 用户手册 (硬件设计 / 维护点检篇)

6.3 CPU 存取密码

(1) 关于 CPU 存取密码

在 CPU 模块中，为了防止来自于误连接的 GX Developer 的非法操作，通过密码进行存取验证。

用于进行存取验证的密码称为 CPU 存取密码。

CPU 存取密码需要在 GX Developer 侧工程及 CPU 模块的双方预先设置。

通过 GX Developer 对 CPU 模块进行控制更改操作（程序更改等）时，CPU 模块将对 GX Developer 侧工程及 CPU 模块的密码进行校验。

只有在校验结果一致时，才允许通过 GX Developer 的操作。

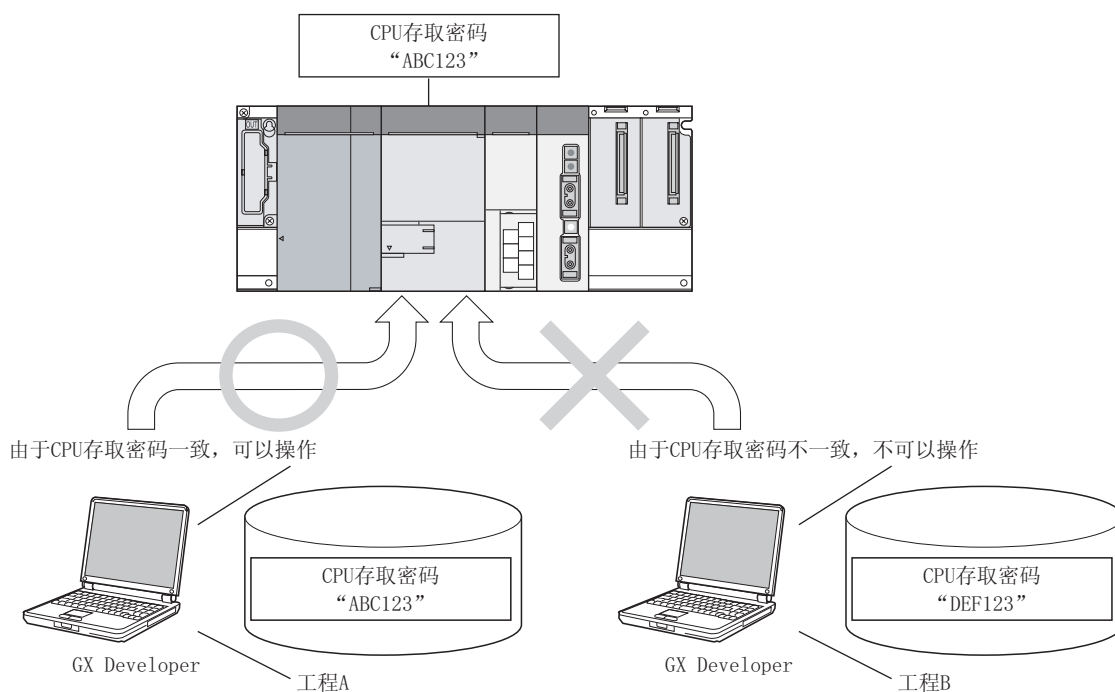


图 6.7 CPU 存取密码

(2) CPU 存取密码的设置及可用字符

(a) CPU 存取密码的设置

CPU 存取密码的设置是通过 GX Developer 的 CPU 存取密码登录 / 更改画面进行的。设置的 CPU 存取密码被登录到工程中。

关于 CPU 存取密码的登录 / 更改操作的详细内容, 请参阅 GX Developer 的手册 (安全可编程控制器篇)。

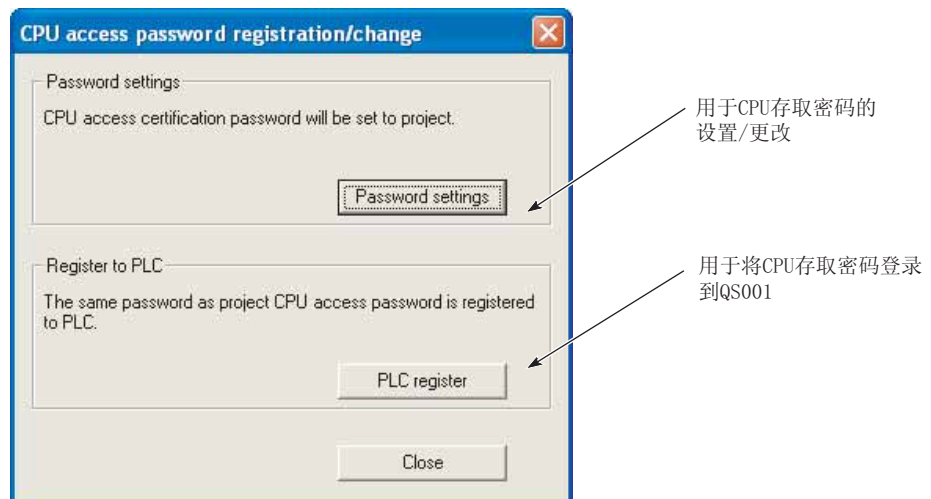


图 6.8 CPU 存取密码登录 / 更改画面

(b) CPU 存取密码中可使用的字符及字符数

CPU 存取密码是用 6 ~ 14 个字符的半角英文、数字及符号 (表 6.7 的带阴影部分) 进行设置。

(区分英文字母的大、小写。)

表 6.7 CPU 存取密码中可使用的字符

MSD		0	1	2	3	4	5	6	7
LSD		000	001	010	011	100	101	110	111
0	0000	NUL	DLE	(SP)	0	@	P	`	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	\$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	{
C	1100	FF	FS	,	<	L	¥	l	
D	1101	CR	GS	-	=	M	]	m	}
E	1110	SO	RS	.	>	N	^	n	~
F	1111	SI	US	/	?	O	_	o	DEL

☒ 要 点

1. 产品出厂时，CPU 模块中未设置 CPU 存取密码。
使用 CPU 模块时，应通过 GX Developer 设置 CPU 存取密码，并登录到 CPU 模块中。
(如果未在 CPU 模块中登录 CPU 存取密码，将不能进行 GX Developer 的在线操作。)
2. CPU 存取密码应由用户妥善管理。
CPU 模块中已经设置了 CPU 存取密码时，如果未在工程侧设置相同的密码，将不能进行可编程控制器写入。
此外，设置的密码也不能更改。
如果忘记了 CPU 存取密码，需要通过可编程控制器存储器初始化对 CPU 模块进行初始化后，重新对工程进行可编程控制器写入。
关于可编程控制器存储器初始化的有关内容，请参阅 6.4 节。
3. 至 CPU 模块的 CPU 存取密码的登录可以在以下场合下进行。
 - CPU 操作模式：测试模式
 - CPU 动作状态：STOP 状态
4. 应对各个 CPU 模块设置不同的 CPU 存取密码。

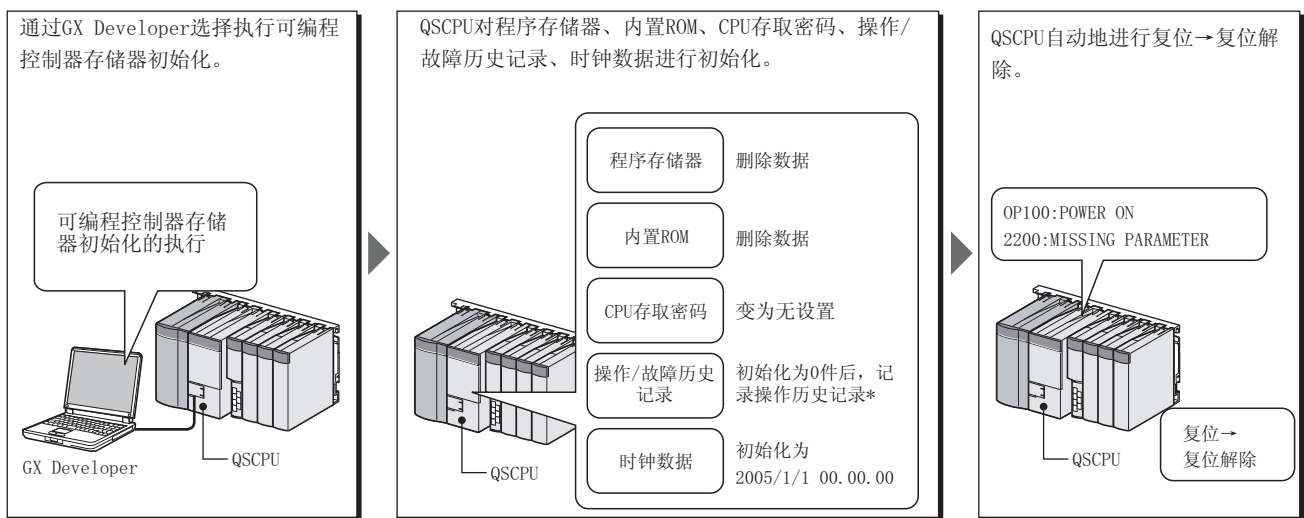
6.4 可编程控制器存储器初始化

(1) 关于可编程控制器存储器初始化

可编程控制器存储器初始化是删除写入到 CPU 模块中的用户数据的功能。

如果执行了可编程控制器存储器初始化，将返回至产品出厂时的状态。

执行可编程控制器存储器初始化后将自动地进行复位→复位解除后，再次进行初始化处理。



*: 操作/故障历史记录中将记录OP005:SYSTEM INITIALIZE PLC MEMORY。

图 6.9 可编程控制器存储器初始化的动作概要

(2) 可编程控制器存储器初始化的处理内容

可编程控制器存储器初始化的处理内容如表 6.8 所示。

表 6.8 可编程控制器存储器初始化的处理内容

项目	初始化的处理内容
程序存储器	数据被删除。(一个文件也不存在的状态)
内置 ROM	数据被删除。(一个文件也不存在的状态)
CPU 存取密码	变为未登录状态。
安全 CPU 动作模式	变为测试模式。
操作 / 故障历史记录	历史记录被删除后，将记录以下操作 / 故障历史记录。 • OP005: SYSTEM INITIALIZE PLC MEMORY • OP100: POWER ON • 2200: MISSING PARAMETER
时钟数据	初始化为 2005/01/01 00:00:00.
ROM 写入次数	加上 2

(3) 可编程控制器存储器初始化的执行可否

可编程控制器存储器初始化在以下的情况下可以执行。

安全 CPOU 动作模式	安全模式		安全模式 (再启动等待)	测试模式	
CPU 动作状态	RUN	STOP	STOP	RUN	STOP
可编程控制器存储器初始化的 执行可否	×	○	○	×	○

○：可以执行； ×：不可执行

(4) 可编程控制器存储器初始化步骤

通过 GX Developer 进行可编程控制器存储器初始化的步骤如图 6.10 所示。

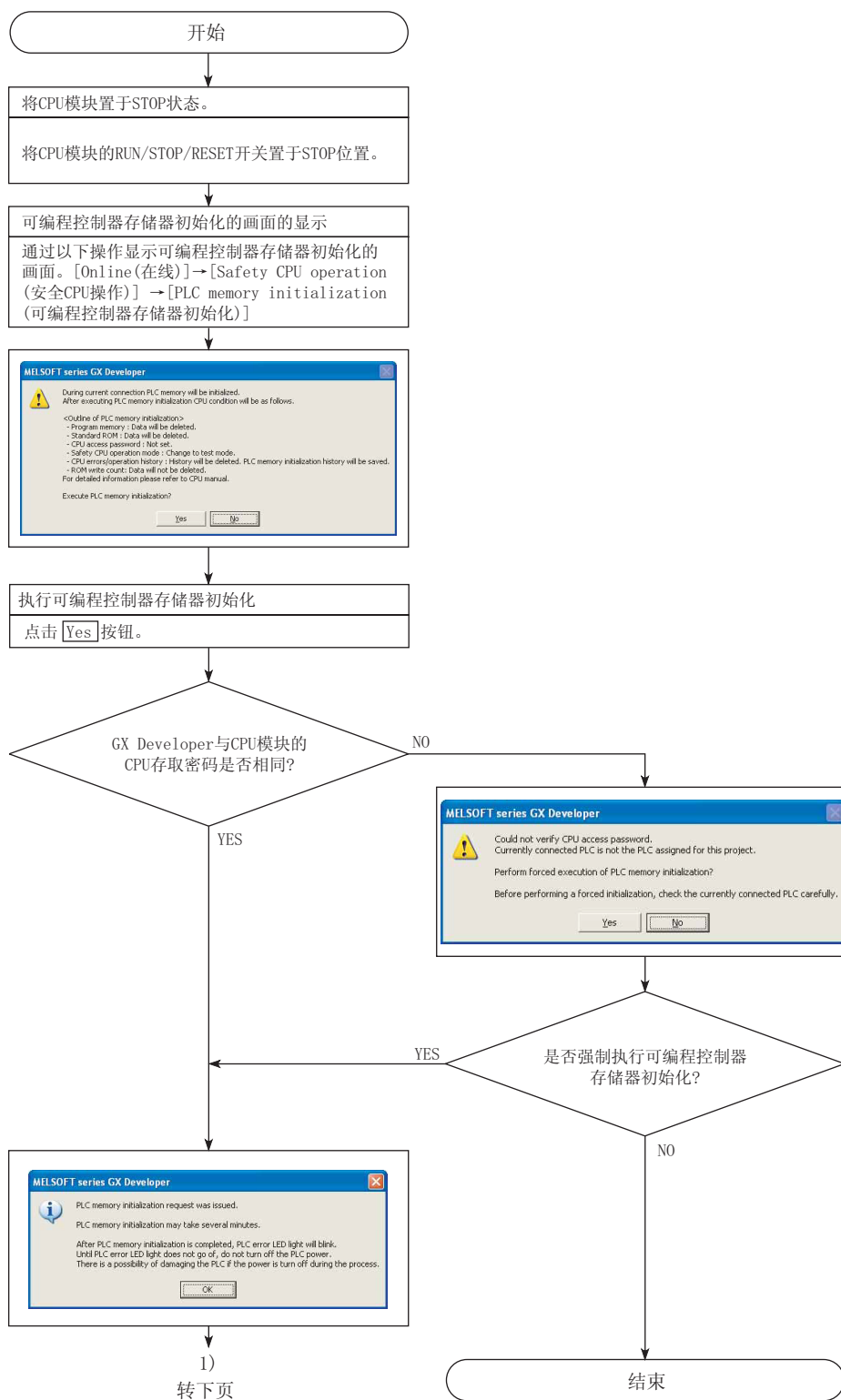


图 6.10 可编程控制器存储器初始化步骤

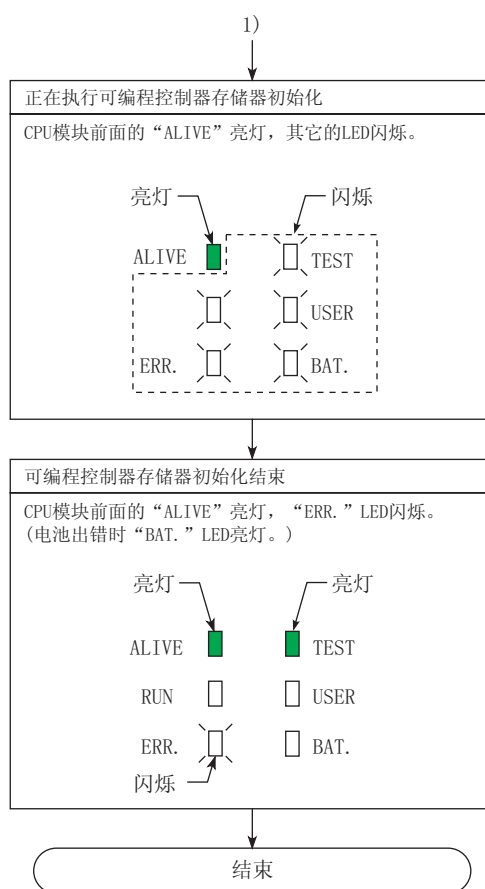


图 6.10 可编程控制器存储器初始化步骤 (续)

(5) 注意事项

(a) CPU 模块出错时的可编程控制器存储器初始化

在 CPU 模块中发生了“INTERNAL CPU COMMUNICATION ERROR”(出错代码:8070 ~ 8074) 等的出错的状态下如果执行可编程控制器存储器初始化, 通过 GX Developer 对 CPU 模块进行写入时有可能会发生通信出错。

应在确认未发生上述出错之后执行可编程控制器存储器初始化。

关于“INTERNAL CPU COMMUNICATION ERROR”等的处理方法, 请参阅 QSCPU 用户手册(硬件设计 / 维护点检篇)中记载的出错代码一览。

(b) 可编程控制器存储器初始化过程中与 GX Developer 的通信

在可编程控制器存储器初始化过程中, 不能通过 GX Developer 对 CPU 模块进行在线操作。

应在可编程控制器存储器初始化结束后, 再过 GX Developer 进行在线操作。

6.5 测试模式下的连续 RUN 的防止设置

(1) 测试模式下的连续 RUN 的防止设置

测试模式下的连续 RUN 的防止设置是为了防止在测试模式下的长时间的连续 RUN。
在测试模式下的 RUN 状态超出了设置的限制时间（测试模式下的连续 RUN 允许时间）时，将发生继续运行出错“TEST MODE TIME EXCEEDED”（出错代码：8100）。

(2) 测试模式下的 RUN 动作连续时间的计量

(a) 计量开始

CPU 模块在测试模式下变为 RUN 状态时，开始计量测试模式下的 RUN 动作连续时间。

(b) 计量终止

CPU 模块变为以下状态时，测试模式下的 RUN 动作连续时间的计量将终止，计量值将被清除。

- 将 CPU 模块置于 STOP 状态时
- 使可编程控制器的电源 OFF 时
- 对 CPU 模块进行复位时

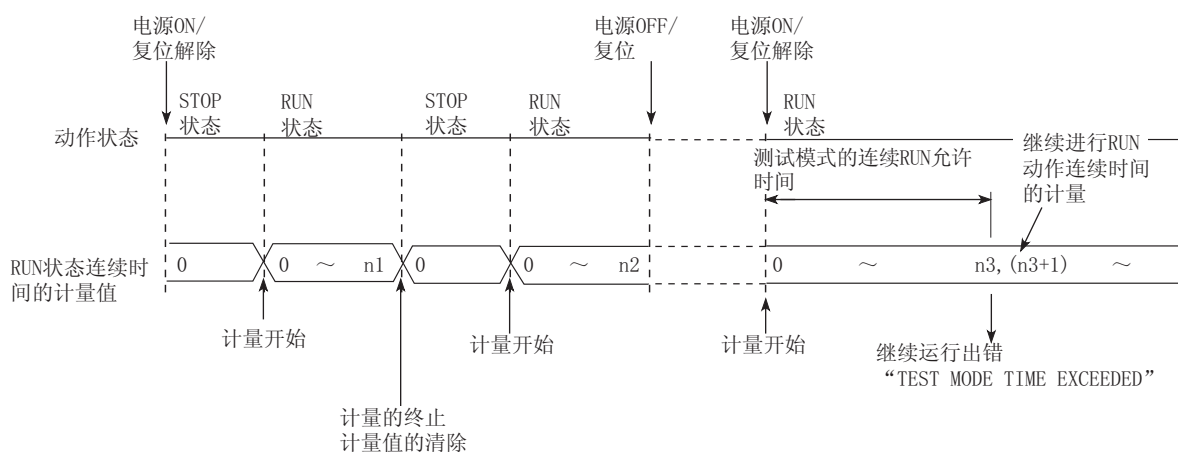


图 6.11 测试模式下的 RUN 动作连续时间的计量期间

备注

1. 在安全模式下的运行过程中，不进行测试模式下的 RUN 动作连续时间的计量。
2. 即使在测试模式下的运行时间超出了设置的测试模式下的连续 RUN 允许时间，发生了继续运行出错“TEST MODE TIME EXCEEDED”（出错代码：8100）的情况下，也继续进行测试模式下的连续 RUN 动作时间的计量。

(3) 测试模式下的连续 RUN 允许时间的设置

测试模式下的连续 RUN 允许时间是在可编程控制器参数的安全设置画面中设置。

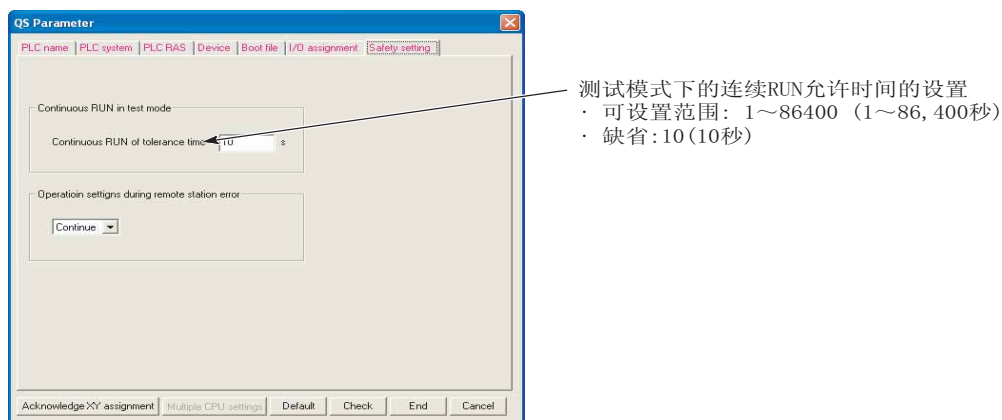


图 6.12 可编程控制器参数的安全设置画面

(4) 测试模式下的 RUN 动作连续时间的确认

测试模式下的 RUN 动作连续时间被存储在特殊寄存器的 SD561、SD562 中。

通过监视特殊寄存器的 SD561、SD562，可以确认测试模式下的 RUN 动作连续时间。

此外，发生了继续运行出错“TEST MODE TIME EXCEEDED”时，特殊继电器的 SM561 将 ON。

表 6.9 存储测试模式下连续 RUN 时间的计量结果的特殊继电器、特殊寄存器

特殊继电器、 特殊寄存器编号	内容	备注
SM561	在测试模式下的 RUN 动作连续时间超出了设置的测试模式下的连续 RUN 允许时间时将 ON。	- 发生变化时更新 - 如果进行出错解除，SM561 将 OFF。
SD561	- 测试模式下的 RUN 动作连续时间是以 BIN 值存储。(单位：秒) - 数据存储范围为 1 ~ 2147483647。	- 每个扫描 END 处理时更新。 - 即使发生了继续运行出错“TEST MODE TIME EXCEEDED”，也将继续进行计量值的存储。
SD562	清除了计量值时 SD561、SD562 也被清除。	如果执行出错解除，SD561、SD562 将被清除。

6.6 至 ROM 的写入次数的确认

至 ROM 的写入次数最多为 10 万次。

如果至 ROM 的写入次数超出了 10 万次，将发生继续运行出错“EXCEED MAX FLASH ROM REWRIT. ERR.”（出错代码：1610）。

CPU 模块发生了“EXCEED MAX FLASH ROM REWRIT. ERR.”（出错代码：1610）时，有可能无法进行至 ROM 的写入，因此应更换 CPU 模块。

(1) 至 ROM 的写入次数的确认方法

至 ROM 的写入次数被存储在特殊寄存器的 SD232、SD233 中。

通过监视 SD232、SD233，可以确认当前的写入次数。

此外，至 ROM 的写入次数超出了 10 万次时，特殊继电器的 SM232 将 ON。

表 6.10 用于确认 ROM 写入次数的特殊继电器、特殊寄存器

特殊继电器、 特殊寄存器编号	内容	备注
SM232	写入次数超出 10 万次时将 ON。	-
SD232	写入次数是以 BIN 值存储。	即使写入次数超出了 10 万次也继续进行写入次数的存储。
SD233		

(2) 被计入 ROM 的写入次数的操作

被计入 ROM 的写入次数的操作如下所示。

(a) 将参数、程序写入内置 ROM

参数、程序的 ROM 化有以下 2 种。

- 通过 GX Developer 进行的程序存储器的 ROM 化
- 从测试模式切换为安全模式时的程序存储器的 ROM 化

(b) 通过 GX Developer 进行的 CPU 存取密码的登录 / 更改

(c) 可编程控制器存储器初始化

☒ 要 点

- 对 ROM 进行写入时的计数值如下所示。
 - 程序存储器的 ROM 化：6
 - 通过 GX Developer 进行的 CPU 存取密码的登录 / 更改：2
 - 可编程控制器存储器初始化：2
- 产品出厂时对 CPU 模块的 ROM 进行了 OS（操作系统）的写入等。
因此 ROM 写入次数为加上了产品出厂时进行的 ROM 写入次数的值。

6.7 自诊断功能

(1) 关于自诊断功能

自诊断功能是 CPU 模块对自身进行有无异常的诊断的功能。

自诊断功能在防止 CPU 模块的误动作的同时还能起到预防维护的目的。

(2) 自诊断的时机

在 CPU 模块的电源接通时或 CPU 模块的运行过程中发生了异常的情况下，通过自诊断功能进行异常检测、出错显示以及 CPU 模块的运算停止等。

(3) 异常的确认

(a) LED 的亮灯

CPU 模块检测出异常时，进行“ERR.”LED 的亮灯等。

(b) 异常内容的存储目标及确认

CPU 模块检测出异常时，特殊继电器 (SM0、SM1) 将 ON，异常内容 (出错代码) 将被存储到特殊寄存器 (SD0) 中。

检测出多个异常时，最新出错的出错代码将被存储到 SD0 中。

应将特殊继电器、特殊寄存器用于程序中，作为可编程控制器或者机械系统的互锁。

(4) 操作 / 故障历史记录的确认为

CPU 模块可以记录 3000 件的操作 / 故障历史记录。

( 6.8 节)

通过 GX Developer 的 [Diagnostics(诊断)] → [PLC diagnostics(可编程控制器诊断)] 可以确认操作 / 故障历史记录。

操作 / 故障历史记录即使是在可编程控制器的电源为 OFF 的情况下也可通过电池进行备份。

(5) 检测出异常时的 CPU 模块的动作 (停止出错 / 继续运行出错)

通过自诊断检测出异常时，CPU 模块的动作有以下 2 种。

(a) 检测出停止 CPU 模块的运算的异常时

在检测出异常时停止 CPU 模块的运算，使外部输出全部为 OFF。(保持软元件存储器的输出 (Y)。)

停止运算的出错称为停止出错。

(b) 检测出 CPU 模块的运算继续进行的出错时

检测出异常后 CPU 模块继续执行程序。

继续进行运算的出错称为继续运行出错。

(6) 自诊断一览

CPU 模块进行的自诊断一览如下所示。

表 6.11 中所示的“出错信息”栏的出错信息可以通过 GX Developer 的 [Diagnostics(诊断)] → [PLC diagnostics(可编程控制器诊断)] 确认。

表 6.11 自诊断一览

No.	详细项目 / 诊断对象	诊断内容	诊断时机	检测出异常时发生的出错	
				出错代码	出错信息
1	RAM 诊断	CPU 模块的内部存储器是否损坏？	电源 ON 时	1131, 1132, 1133,	RAM ERROR
			复位解除时	1136, 1137	
			常时	1141, 1142, 1143, 1146	
2	F/W 诊断	ROM 中存储的 F/W 是否损坏？	电源 ON 时 复位解除时 执行 END 指令时	8060	INCORRECT FIRMWARE
3	运算回路诊断	CPU 模块内的执行顺控程序运算的运算回路是否正常动作？	电源 ON 时 复位解除时 执行 END 指令时	1210	OPERATION CIRCUIT ERROR
4	文件校验	程序存储器中存储的文件是否损坏？	电源 ON 时 复位解除时	8031	INCORRECT FILE
			执行 END 指令时	8032	
5	输出数据校验	CPU A/B 各自运算的输出结果是否一致？	执行 END 指令时	8050	SAFETY OUTPUT VERIFY ERROR
6	时间的监视	CPU A/B 各自的 OS(操作系统) 的执行状态是否相同？	常时	8020	CPU A & B CAN'T BE SYNCHRONIZED
			执行 END 指令时	8021	
7	微机诊断	CPU 模块中使用的寄存器是否正常动作？	电源 ON 时 复位解除时 执行 END 指令时	8000	INTERNAL REGISTER ERROR
				8010	INTERNAL BUS ERROR
8	电源电压监视	CPU 模块是否在动作保障范围内的电压下运行？	常时	8080	POWER SUPPLY ERROR
9	电源电压监视回路诊断	监视电源电压的回路是否正常动作？	执行 END 指令时	8090	VOLTAGE DIAGNOSIS ERROR
10	检测出时钟停止	至 CPU 模块内部回路的输入时钟是否停止？	常时	8120	WDT CLOCK CHECK ERROR
11	CPU 模块的 OS(操作系统)	主 CPU 是否在未失控的状态正常动作？	常时	1000, 1006	MAIN CPU DOWN
12	CPU 模块的硬件	CPU 模块的下述硬件是否正常动作？ • 主 CPU • 时钟单元 • RUN/STOP/RESET 开关	常时	1001, 1002, 1003, 1004	MAIN CPU DOWN
13	电源模块	电源模块是否故障？	常时	1009	MAIN CPU DOWN
14	程序	用户程序的最后是否执行了 END 指令？	执行 END 指令时	1010	END NOT EXECUTE
15	CPU 模块、基板、CC-Link Safety 主站模块、MELSECNET/H 模块	CC-Link Safety 主站模块、MELSECNET/H 模块、基板、CPU 模块内部是否发生了非法中断？	发生中断时	1311	I/O INTERRUPT ERROR
16	运行中的模块状态	CC-Link Safety 主站模块、MELSECNET/H 模块是否正常动作？	电源 ON 时 复位解除时	1401	INTELLIGENT FUNCTION MODULE DOWN
			执行 END 指令时	1403	
17	CC-Link Safety 主站模块、MELSECNET/H 模块的通信路径	• 是否能与 CC-Link Safety 主站模块、MELSECNET/H 模块正常通信？ • 基板是否正常动作？	电源 ON 时 复位解除时	1411	CONTROL-BUS ERROR
			常时	1413	
			执行 END 指令时	1414, 1415	
18	电源模块的输入电源	• 电源模块的输入电源是否正常供应？ • 电源模块的输入电源是否发生了瞬间掉电？	常时	1500	AC DOWN

(转下页)

表 6.11 自诊断一览 (续)

No.	详细项目 / 诊断对象	诊断内容	诊断时机	检测出异常时发生的出错	
				出错代码	出错信息
19	电池	CPU 模块中安装的电池电压是否在规定值以上 ?	常时	1600	BATTERY ERROR
20	ROM 写入次数	至 ROM 的写入次数是否在保证次数 (10 万次) 以内 ?	执行 END 指令时	1610	EXCEED MAX FLASH ROM REWRIT. ERR.
21	运行中的模块安装状态	电源 ON/ 复位解除之后 CC-Link Safety 主站模块、MELSECNET/H 模块的安装状态有无变化 ?	执行 END 指令时	2000	MODULE VERIFY ERROR
22	模块构成	- CC-Link Safety 主站模块、MELSECNET/H 模块的安装状态是否与可编程控制器参数的 I/O 分配设置相符 ? - CC-Link Safety 主站模块、MELSECNET/H 模块的安装个数是否超过了允许安装个数 ? - CC-Link Safety 主站模块、MELSECNET/H 模块的起始 I/O 地址号有无重复 ?	电源 ON 时 复位解除时	2100, 2106, 2107	MODULE LAYOUT ERROR
		实际 I/O 点数以后是否安装了模块 ?	电源 ON 时 复位解除时	2124	MODULE LAYOUT ERROR
		是否安装了不能使用的模块 (I/O 模块、智能功能模块、GOT) ?	电源 ON 时 复位解除时	2125	MODULE LAYOUT ERROR
23	参数构成	CPU 模块内是否存在有参数 ?	电源 ON 时 复位解除时	2200	MISSING PARAMETER
24	参数设置	可编程控制器参数的设置是否符合规格 ?	电源 ON 时 复位解除时	3000, 3001, 3003, 3004, 3008	PARAMETER ERROR
		MELSECNET/H 的网络参数的设置是否符合规格 ?	电源 ON 时 复位解除时	3100, 3101 3102, 3104	NETWORK PARAMETER ERROR
		CC-Link Safety 参数的设置是否符合规格 ?	电源 ON 时 复位解除时	3105, 3106 3107	CC-LINK PARAMETER ERROR
25	程序	程序的指令代码是否正确 ? (程序的指令代码是否已损坏 ?)	电源 ON 时 复位解除时 STOP → RUN 时	4000	INSTRUCTION CODE ERROR
		程序的扩展指令 (S.QSABORT 指令) 的格式是否正确 ?	电源 ON 时 复位解除时 STOP → RUN 时	4002, 4003, 4004	INSTRUCTION CODE ERROR
		程序中是否存在有 END 指令 ?	电源 ON 时 复位解除时 STOP → RUN 时	4010	MISSING END INSTRUCTION
		执行指令时, 嵌入到指令中的输入数据是否符合指令规格 ?	执行指令时	4100, 4101	OPERATION ERROR
26	扫描时间	扫描时间是否在 WDT 的时限设置值以内 ?	常时	5001	WDT ERROR
		恒定扫描时间设置时, 1 个扫描能否在恒定扫描时间以内结束 ?	常时	5010	PROGRAM SCAN TIME OVER
27	测试模式下的运行时间	测试模式下的连续运行时间是否在限制值以内 ?	执行 END 指令时	8100	TEST MODE TIME EXCEEDED

6.7.1 出错时的 LED 显示

发生出错时，CPU 模块前面的 LED 将亮灯或闪烁。(☞ 6.17 节)

6.7.2 出错的解除

对于继续进行程序运算的出错，CPU 模块可以通过程序进行出错的解除操作。
对于发生的继续运行出错，可以通过 SD81(出错原因) 的位是否变为 1 来确认。与 SD81 的位号相对应的出错原因、继续运行出错如表 6.12 所示。

表 6.12 与 SD81 的位号相对应的出错原因、出错代码

对应于继续运行出错的 SD81 的位号及出错原因		对应于 SD81 的位号的继续运行出错	
位号	出错原因	出错代码	出错信息
0	发生瞬时掉电	1500	AC/DC DOWN
1	电池电压过低	1600	BATTERY ERROR
2	内置 ROM 写入次数溢出	1610	EXCEED MAX FLASH ROM REWRIT.ERR.
3	测试模式连续 RUN 允许时间溢出	8100	TEST MODE TIME EXCEEDED
4	扫描时间溢出	5010	PROGRAM SCAN TIME OVER
5	报警器 ON	9000	F**** (**** 为报警器号。)
6	安全远程站检测出出错	8300	CC-LINK REMOTE DETECTION ERROR
7	安全远程站产品信息不一致	8310	CC-LINK PRODUCT INFO. MISMATCH
8	初始化监视超时出错	8320	CC-LINK DATA RECEPTION TIMEOUT
	安全监视超时出错	8321	
	出错监视超时出错	8322	
9	安全远程站指令异常	8330	CC-LINK RECEIVED DATA ERROR
	安全远程站数据分割异常	8331	
	安全远程站链接 ID 异常	8332	
	安全远程站顺序号异常	8333	
	安全远程站接收数据异常	8334	

(1) 出错解除步骤

出错的解除步骤如下所示。

- 1) 通过 GX Developer 读取 SD81，确认当前 CPU 模块中发生的继续运行出错的原因。
- 2) 消除出错的原因。
- 3) 特殊寄存器 SD50 中将存储解除的出错代码。
- 4) 使特殊继电器 SM50 由 OFF → ON。
- 5) 再次通过读取 SD81，确认解除的继续运行出错相对应的位是否为 OFF。
- 6) 使特殊继电器 SM50 为 OFF。

(a) 多个出错的出错解除步骤

由于当最新的出错（存储在特殊寄存器 SD0 中的出错）被解除时，出错信息特殊继电器 / 特殊寄存器（SM0、SM1、SM5、SM16、SD0 ~ 26）的内容将被清除，未被解除的出错的信息将不能从特殊继电器 / 特殊寄存器中获取。

应通过出错历史记录获取以前发生的未被解除的出错后，将其解除。

(☞ 6.8 节)。

(2) 出错解除后的状态

通过出错解除使 CPU 模块复原时，与出错相关的特殊继电器、特殊寄存器以及 LED 将恢复为出错前的状态。

故障历史记录不留下痕迹。

进行了出错解除后再次发生了相同的出错时，将被再次登录到操作 / 故障历史记录中。

(3) 报警器的解除

解除多个检测出的报警时，只能解除最初检测出的 F 编号。

(4) 发生了多个继续运行出错时的出错解除

进行发生了多个继续运行出错时的出错解除的情况下，CPU 模块的 LED 显示、出错信息的动作如下所示。

出错解除状态	LED 显示 *1 (ERR. LED, BAT. LED, USER LED)	出错信息 (SM0, SM15, SM16, SD0 - 26)
出错解除前	亮灯	存储最后发生的继续运行出错的出错信息。
↓		
解除了最后发生的继续运行出错。 (剩余有未解除的继续运行出错。)	亮灯	恢复为无出错的状态。
解除了发生的继续运行出错以外。 (剩余有未解除的继续运行出错。)	亮灯	无变化。 (保持最后发生的出错信息。)
↓		
解除了所有的继续运行出错。	熄灯	无出错。

*1: (1) 发生了出错代码 :1600(“BATTERY ERROR”)的出错时，仅“BAT.”LED 亮灯。

解除了出错代码 :1600 的出错后，“BAT.”LED 将熄灯。

(2) 发生了出错代码 :9000(F****)的出错时，仅“USER”LED 亮灯。

解除了出错代码 :9000 的出错后，“USER”LED 将熄灯。

☒ 要 点

1. 将解除的出错代码存储到 SD50 中进行了出错解除时，低 1 位的代码编号将被忽略。

(例)

发生了出错代码 2100、2106 的出错时，如果解除了出错代码 2100 的出错，出错代码 2106 也将被解除。

发生了出错代码 2100、2125 的出错时，即使解除了出错代码 2100 的出错，也不能解除出错代码 2125 的出错。

2. 对于 CPU 模块以外的原因导致发生的出错，即使通过特殊继电器 (SM50) 以及特殊寄存器 (SD50) 进行了出错解除，也不能消除出错原因。

(例)

由于“INTELLIGENT FUNCTION MODULE DOWN”是基板、智能功能模块等中发生的出错，因此即使通过特殊继电器 (SM50) 以及特殊寄存器 (SD50) 进行了出错解除，也不能消除出错原因。

请参阅 QSCPU 用户手册（硬件设计 / 维护点检篇）中记载的出错代码一览，消除出错原因。

6.8 操作内容、自诊断出错发生内容的记录（操作 / 故障历史记录功能）

(1) 关于操作 / 故障历史记录

操作 / 故障历史记录功能是指，通过记录从外部对 CPU 模块执行的操作及 CPU 模块中以前发生的自诊断出错，使故障排除易于进行的功能。

(2) 操作 / 故障历史记录区中存储的数据

CPU 模块将从外部对 CPU 模块执行的操作及自诊断出错存储到操作 / 故障历史记录区中。

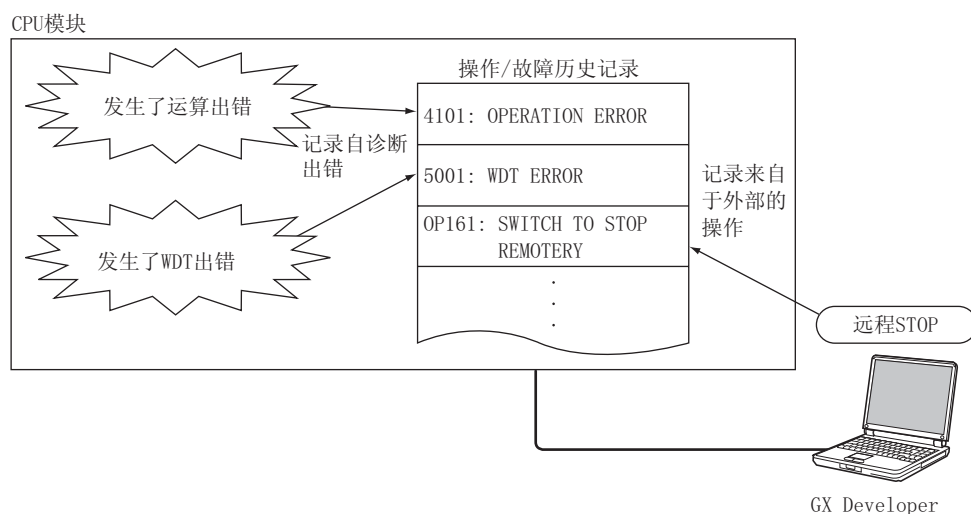


图 6.13 CPU 模块中记录的操作 / 故障历史记录

(a) 从外部对 CPU 模块执行的操作

以下操作将被作为从外部对 CPU 模块执行的操作而被存储。

- 通过 GX Developer 进行的在线操作
- 通过 CPU 模块的 RUN/STOP/RESET 开关进行的操作
- 输入电源的 ON/OFF

操作 / 故障历史记录中存储的操作如表 6.13 所示。

表 6.13 操作 / 故障历史记录中存储的操作

分类	操作代码	操作信息	操作说明
系统	OP001	SYSTEM INITIALIZE OPERATION MODE	由于未能正确地保持安全 CPU 动作模式，因此 CPU 模块将安全 CPU 动作模式初始化为测试模式。
	OP002	SYSTEM INITIALIZE PROGRAM MEMORY	由于未能正确地保存程序存储器的内容，因此 CPU 模块对程序存储器进行了格式化。
	OP003	SYSTEM INITIALIZE OPE./ERROR LOG	由于未能正确地保存操作 / 故障历史记录的内容，因此操作 / 故障历史记录被初始化为 0 件。
	OP004	SYSTEM INITIALIZE SYSTEM CLOCK	由于系统的时钟数据不正确，因此 CPU 模块对系统的时钟数据进行了初始化。
	OP005	SYSTEM INITIALIZE PLC MEMORY	CPU 模块执行了可编程控制器存储器初始化功能。
	OP006	SYSTEM INITIALIZE ROM WRITE INF.	由于未能正确地保存 ROM 化信息，因此 CPU 模块对 ROM 化信息进行了初始化。
系统 (CPU 动作状态)	OP010	SYSTEM SWITCH TO RUN	CPU 模块的 CPU 动作状态被切换为 RUN 状态。
	OP011	SYSTEM SWITCH TO STOP	CPU 模块的 CPU 动作状态被切换为 STOP 状态。
电源操作	OP100	POWER ON	进行了可编程控制器电源的 ON 操作。或者对 CPU 模块进行了复位解除。
驱动器操作	OP144	WRITE PROGRAM MEMORY TO ROM	进行了程序存储器→内置 ROM 的 ROM 化。
远程操作	OP160	SWITCH TO RUN REMOTELY	进行了远程 RUN 操作。
	OP161	SWITCH TO STOP REMOTELY	进行了远程 STOP 操作。
安全 CPU 动作模式操作	OP180	SWITCH SAFETY PC OPERATION MODE	对安全 CPU 动作模式进行了切换。
历史记录操作	OP200	CLEAR OPERATION/ERROR LOG	对 CPU 模块内的操作 / 故障历史记录进行了清除。
时钟操作	OP210	ADJUST SYSTEM CLOCK	进行了 CPU 模块的时钟设置。
CPU 存取密码操作	OP220	MODIFY ACCESS PASSWORD	对 CPU 模块的 CPU 存取密码进行了设置。

(b) 自诊断出错

CPU 模块将存储检测出的自诊断出错的内容。

关于自诊断出错的详细内容请参阅以下手册。

☞ QSCPU 用户手册 (硬件设计 / 维护点检篇)

(3) 操作 / 故障历史记录的容量

CPU 模块的操作 / 故障历史记录中可以存储 3000 件的操作内容、出错内容。

当操作内容与出错内容的合计超出了 3000 件时，将从最旧的记录内容开始按顺序被覆盖。

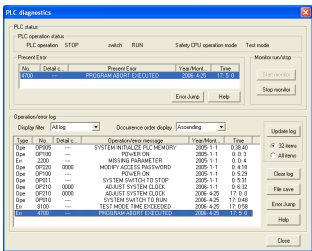
(4) 通过 GX Developer 显示操作 / 故障历史记录

操作 / 故障历史记录的内容可以通过 GX Developer 的可编程控制器诊断画面显示。

(a) 可编程控制器诊断画面的显示

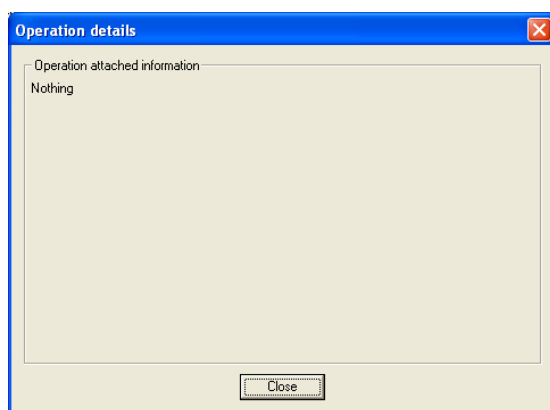
GX Developer 的可编程控制器诊断画面中显示的操作 / 故障历史记录如表 6.14 所示。

表 6.14 可编程控制器诊断画面及操作 / 故障历史记录项目的内容

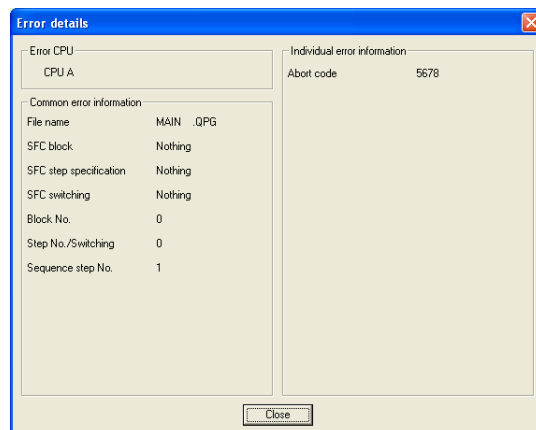
可编程控制器诊断画面	可编程控制器诊断画面的说明	
	项目	说明
	类型	显示历史记录的类型。 Ope: 操作历史记录 Err: 出错历史记录
	No.	显示操作 / 出错代码。
	详细代码	显示操作历史记录以及 CC-Link Safety 远程 I/O 模块的出错历史记录对应的 4 位代码。 无详细代码时, 将显示 ----。
	当前的出错 / 出错信息	显示操作 / 故障历史记录中记录的操作内容、出错信息。 当历史记录被损坏时, 将显示 “BROKEN OPERATION/ERROR LOG”。

(b) 操作 / 故障历史记录の詳細画面

如果在可编程控制器诊断画面中用鼠标双击当前发生的出错或者历史记录一览内的历史记录, 可以显示图 6.14 的详细画面。



(a) 操作历史记录



(b) 故障历史记录

图 6.14 操作历史记录 / 故障历史记录の詳細画面

(5) 操作 / 故障历史记录清除

通过在 GX Developer 的可编程控制器诊断画面中点击 “Clear log(历史记录清除)” 按钮，可以清除 CPU 模块的操作 / 故障历史记录。

操作 / 故障历史记录的清除操作只有在 CPU 模块的安全 CPU 动作模式为测试模式时才有效。如果清除操作 / 故障历史记录，CPU 模块将在操作 / 故障历史记录中存储操作内容 OP200: “CLEAR OPERATION/ERROR LOG”。

☒ 要 点

操作 / 故障历史记录是通过 CPU 模块的电池保存。

CPU 模块在进行了电源 ON/ 复位解除时将确认操作 / 故障历史记录是否丢失或被破坏。

如果检测出由于电池电量的过低等导致操作 / 故障历史记录数据丢失或者被破坏时，CPU 模块将对操作 / 故障历史记录进行初始化。

CPU 模块对操作 / 故障历史记录进行了初始化时，操作 / 故障历史记录中将存储操作内容 OP003: “SYSTEM INITIALIZE OPE./ERROR LOG”。

6.9 恒定扫描

(1) 关于恒定扫描

由于扫描时间根据顺控程序中使用的指令的执行与否其处理时间也不一样，因此每次扫描均不一样。

恒定扫描是将扫描时间保持为一定的时间的状况下反复执行顺控程序的功能。

(2) 恒定扫描的用途

I/O 刷新是在执行顺控程序之前进行的。

通过使用恒定扫描功能，即使顺控程序的执行时间变化，也可以使 I/O 刷新的间隔保持为一定值。

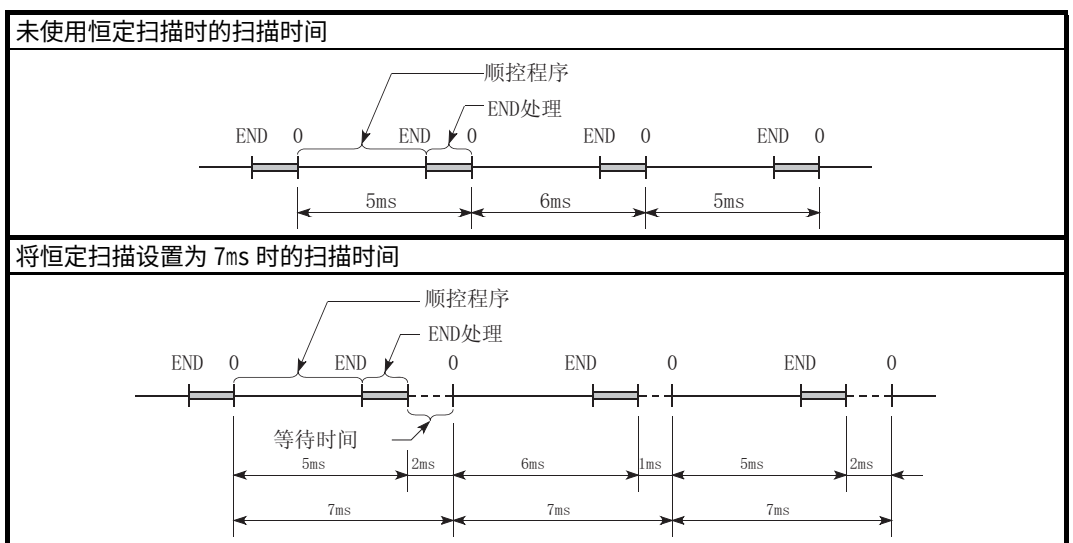


图 6.15 恒定扫描的动作

(3) 恒定扫描时间的设置

恒定扫描时间的设置是通过可编程控制器参数的 PC PAS 设置进行的。

恒定扫描时间可以在 1 ~ 2000ms(设置单位 :1 ms) 的范围内设置。

执行恒定扫描时, 设置恒定扫描时间。

不执行恒定扫描时, 使恒定扫描时间为 “blank(空白)”。

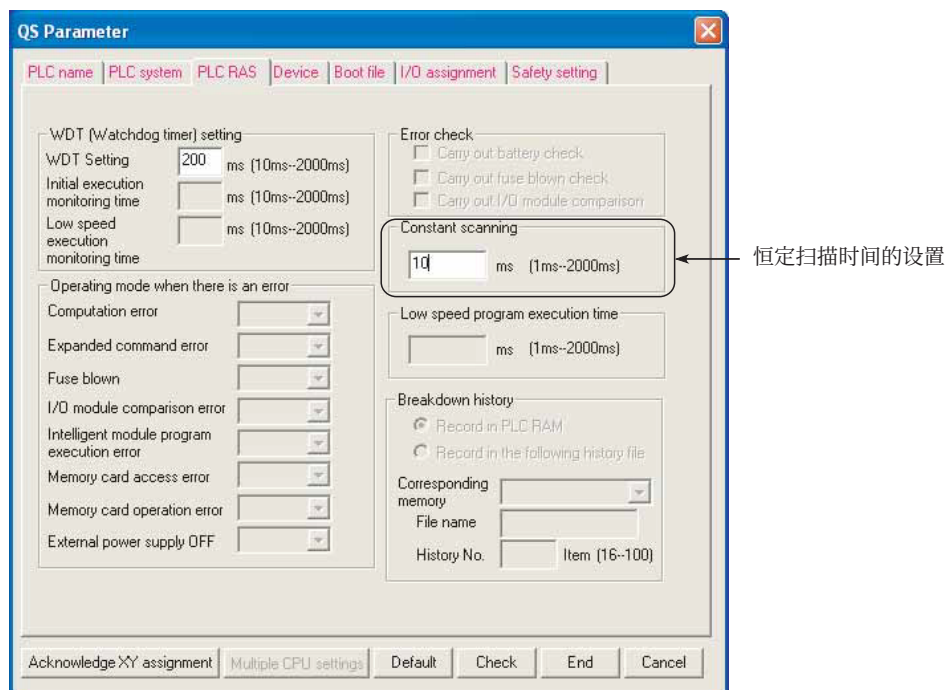


图 6.16 将恒定扫描时间设置为 10ms 时

(a) 设置时间的条件

恒定扫描的设置时间值应满足以下关系式。

$$\begin{aligned} &(\text{WDT 的设置时间}) > (\text{恒定扫描的设置时间}) \\ &> (\text{顺控程序的最长扫描时间}) \end{aligned}$$

顺控程序的扫描时间长于恒定扫描的设置时间时，CPU 模块将检测出“PROGRAM SCAN TIME OVER”(出错代码:5010)。

在这种情况下，将忽略恒定扫描时间而执行顺控程序的扫描时间。

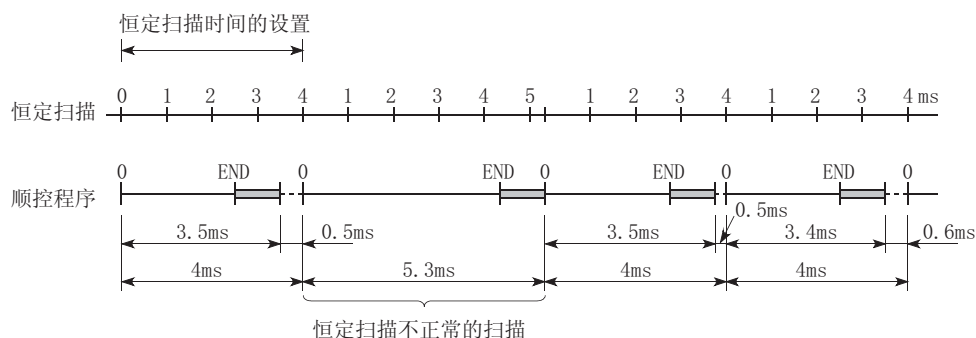


图 6.17 扫描时间长于恒定扫描的设置时间时的动作

顺控程序的扫描时间长于 WDT 的设置时间时，CPU 模块将检测出 WDT 出错。

在这种情况下，将停止执行程序。

(4) 关于从执行 END 处理起至下一个扫描开始为止的等待时间

在执行顺控程序的 END 处理后，至下一个扫描开始为止的等待期间，顺控程序的处理将中止。

(5) 恒定扫描的精度

关于恒定扫描的精度，请参阅第 10 章。

6.10 STOP 状态 $\longleftrightarrow$ RUN 状态时的输出 (Y) 状态的设置

(1) CPU 模块从 RUN 状态变为 STOP 状态时，将 RUN 状态的输出 (Y) 记忆到可编程控制器内部，将输出 (Y) 全部 OFF。

在 CPU 模块中，从 STOP 状态变为 RUN 状态时的状态可以从以下的 2 个中选择。

- 输出 STOP 前的输出 (Y) 状态
- 清除输出 (Y)。

(2) 设置用途

通过保持回路等，可以对 STOP 状态 $\rightarrow$ RUN 状态时的输出选择是否从上一次的状态重启。



图 6.18 保持回路

- 将 STOP 前的输出 (Y) 状态设置为输出时

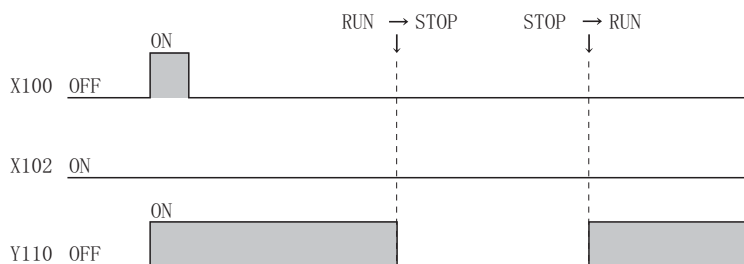


图 6.19 将 STOP 前的输出 (Y) 状态设置为输出时的时序图

- 将输出 (Y) 设置为清除时

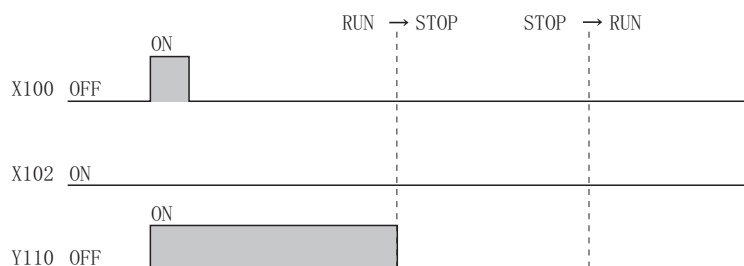


图 6.20 将输出 (Y) 设置为清除时的时序图

(3) 从 STOP 状态变为 RUN 状态时的动作

- (a) 输出 STOP 前的输出 (Y) 状态 (缺省)
输出了变为 STOP 状态之前的输出 (Y) 状态后, 进行顺控程序的运算。
- (b) 清除输出 (Y) (输出 1 个扫描后)
输出变为 OFF 状态。
输出 (Y) 的输出是在执行了顺控程序的运算后进行。
关于 STOP 状态时将输出 (Y) 强制 ON 时的动作, 请参阅本节 (5)。

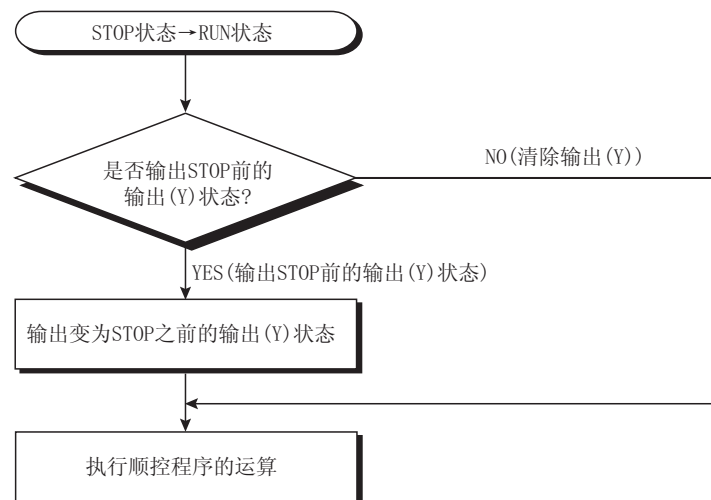


图 6.21 STOP 状态→RUN 状态时的处理

(4) 从 STOP 状态变为 RUN 状态时的输出 (Y) 状态的设置

STOP 状态变为 RUN 状态时的输出 (Y) 状态的设置是在可编程控制器参数的可编程控制器系统设置中进行。

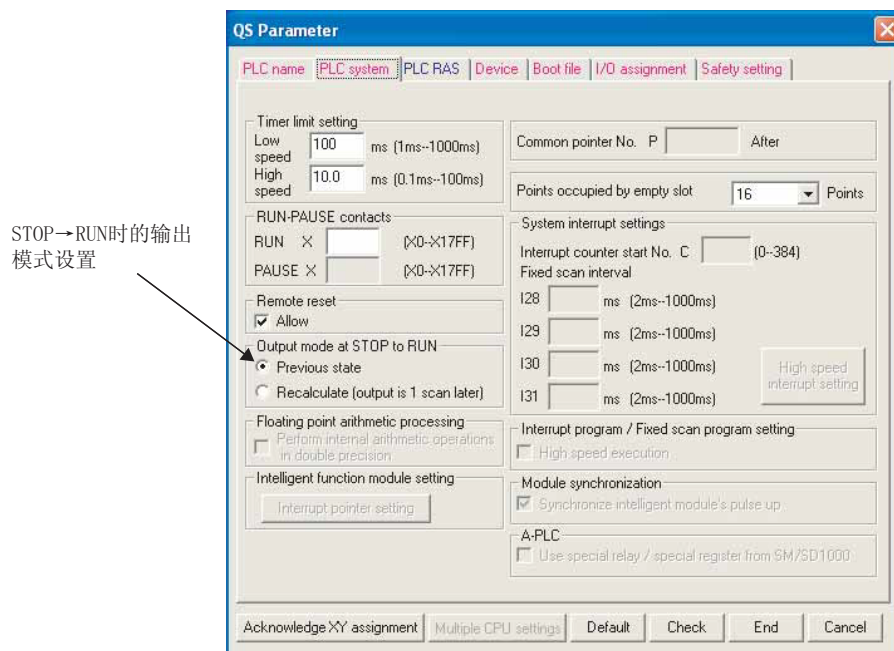


图 6.22 可编程控制器系统设置画面

(5) 注意事项

CPU 模块将 STOP 状态时的输出 (Y) 强制 ON 时，从 STOP 状态变为 RUN 状态时的输出如表 6.15 所示。

表 6.15 STOP 状态时将输出 (Y) 强制 ON 后，从 STOP 状态变为 RUN 状态时的输出

STOP → RUN 时的输出模式	从 STOP 状态变为 RUN 状态时的输出
输出 STOP 前的输出 (Y) 状态	输出 STOP 前的状态。 如果 STOP 前输出 (Y) 为 OFF，将不保持 ON 状态。
清除输出 (Y)	保持 ON 状态。

6.11 时钟功能

(1) 关于时钟功能

时钟功能是指，通过顺控程序读取 CPU 模块内部的时钟数据，用于时间管理的功能。通过将日期存储到操作 / 故障历史记录等，时钟数据被用于 CPU 模块的系统执行功能的时间管理等。

(2) 电源 OFF 以及瞬时掉电时的时钟动作

即使是在可编程控制器的电源 OFF 或者发生了超出允许瞬时掉电时间的停电时，时钟动作也可通过 CPU 模块的内部电池而继续进行。

(3) 时钟数据

时钟数据是 CPU 模块内部使用的时钟数据，其内容如表 6.16 所示。

表 6.16 时钟数据的内容

数据名称	内容	
年	公历 4 位 (可以计测 1980 年 ~ 2079 年)	
月	1 ~ 12	
日	1 ~ 31 (自动判别闰年)	
时	0 ~ 23 (24 小时制)	
分	0 ~ 59	
秒	0 ~ 59	
星期	0	星期日
	1	星期一
	2	星期二
	3	星期三
	4	星期四
	5	星期五
	6	星期六

(4) 时钟数据的更改及读取

(a) 时钟数据的更改

时钟数据的更改是通过 GX Developer 进行的。

通过 GX Developer 的 [Online(在线)] → [Set clock(时钟设置)] 显示时钟设置窗口，更改 CPU 模块的时钟数据。

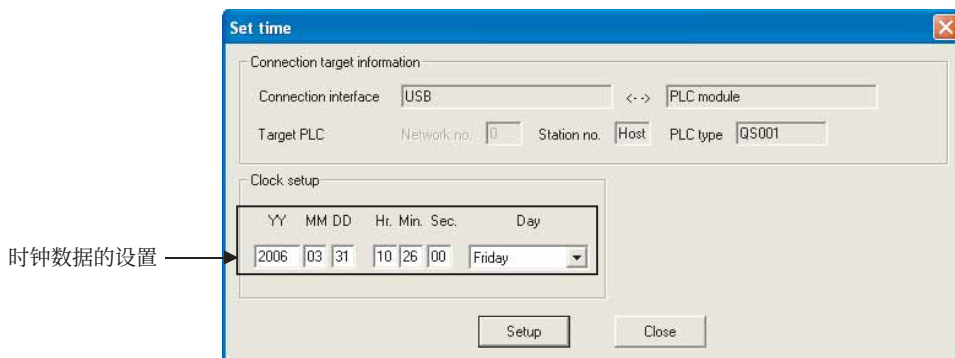


图 6.23 通过 GX Developer 写入时钟数据

☒ 要 点

通过 GX Developer 更改 CPU 模块的时钟数据后，CPU 模块在操作 / 故障历史记录中将记录 OP210: “ADJUST SYSTEM CLOCK”。

(b) 时钟数据的读取

时钟数据的读取是在特殊继电器 (SM213) 及特殊寄存器 (SD210 ~ 213) 中进行的。特殊继电器的详细内容请参阅附录 1，特殊寄存器的详细内容请参阅附录 2。

(5) 注意事项

(a) 初次使用时的时钟数据设置

在产品出厂时未进行时钟数据设置。

时钟数据被用于操作 / 故障历史记录等 CPU 模块的系统。

初次使用 CPU 模块时，必须正确地进行时间设置。

(b) 电池电压过低时的时钟数据

CPU 模块在停电状态下通过使用 CPU 模块中安装的电池进行时钟计测。

因此，CPU 模块的电池容量过低时，时钟数据的值将可能变为不定值。

CPU 模块在可编程控制器的电源 ON/CPU 模块的复位解除时，可以确认时钟数据的值是否在本节 (3) 中所示的范围内。

当时钟数据的值不正确时，时钟数据的值将被初始化为 2005 年 1 月 1 日 00 时 00 分 00 秒。

此时，CPU 模块将在操作 / 故障历史记录中记录 OP004: “SYSTEM INITIALIZE SYSTEM CLOCK”。

(操作 / 故障历史记录中记录的时间使用时钟数据初始化后的时钟数据的值。)

(6) 时钟数据的精度

时钟功能的精度随着环境温度的变化而有所不同，其情况如下所示。

表 6.17 时钟数据的精度

环境温度 (°C)	精度 (日差, s)
0	- 3.18 ~ + 5.25 (TYP. + 2.14)
+ 25	- 3.18 ~ + 5.29 (TYP. + 2.07)
+ 55	- 12.97 ~ + 3.63 (TYP. -3.16)

1

概要

2

性能规格

3

顺控程序的执行

4

I/O 地址号的分配

5

关于 CPU 模块中使用的存储器及文件

6

功能

7

与智能功能模块的通信

8

参数

6.12 远程操作

远程操作是指，通过外部（GX Developer、RUN 触点）操作更改 CPU 模块的动作状态的操作。

远程操作有以下 2 种：

- 远程 RUN/STOP :  6.12.1 节
- 远程 RESET :  6.12.2 节

6.12.1 远程 RUN/STOP

(1) 关于远程 RUN/STOP

远程 RUN/STOP 是指，在将 CPU 模块的 RUN/STOP/RESET 开关置于 RUN 位置不变的情况下从外部进行 CPU 模块的 RUN/STOP 操作。

(2) 关于远程 RUN/STOP 的用途

在以下的情况下，通过远程操作进行远程 RUN/STOP 将带来方便。

- CPU 模块位于较远位置时
- 通过外部信号对控制盘内的 CPU 模块进行 RUN/STOP 时。

(3) 远程 RUN/STOP 时的运算

进行远程 RUN/STOP 的程序的运算情况如下所示。

(a) 远程 STOP

程序执行至 END 指令为止，变为 STOP 状态。

(b) 远程 RUN

将通过远程 STOP 进入 STOP 状态的运算通过执行远程 RUN 再次进入 RUN 状态，从步 0 开始执行程序。

(4) 远程 RUN/STOP 方法

远程 RUN/STOP 方法中，有通过 RUN 触点的方法及通过 GX Developer 的方法这 2 种。

(a) 通过 RUN 触点的方法

RUN 触点是在可编程控制器参数的可编程控制器系统设置中进行设置。

可设置的范围为输入 X0 ~ 17FF。

通过设置的 RUN 触点的 ON/OFF，可以进行远程 RUN/STOP。

- RUN 触点为 OFF 时，CPU 模块变为 RUN 状态。
- RUN 触点为 ON 时，CPU 模块变为 STOP 状态。

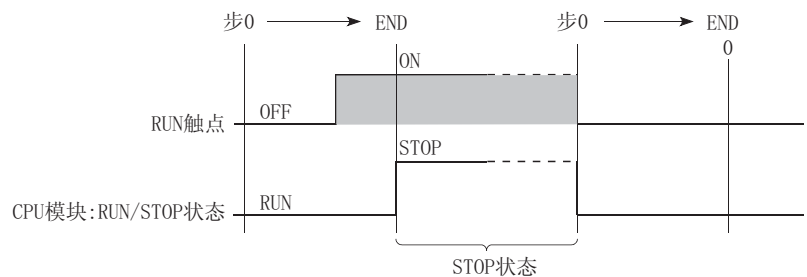


图 6.24 通过 RUN 触点进行 RUN/STOP

(b) 通过 GX Developer 的方法

通过 GX Developer 的远程 RUN/STOP 操作，可以执行 CPU 模块的 RUN/STOP。

GX Developer 的操作是通过 [Online(在线)] → [Remote operation(远程操作)] 进行。

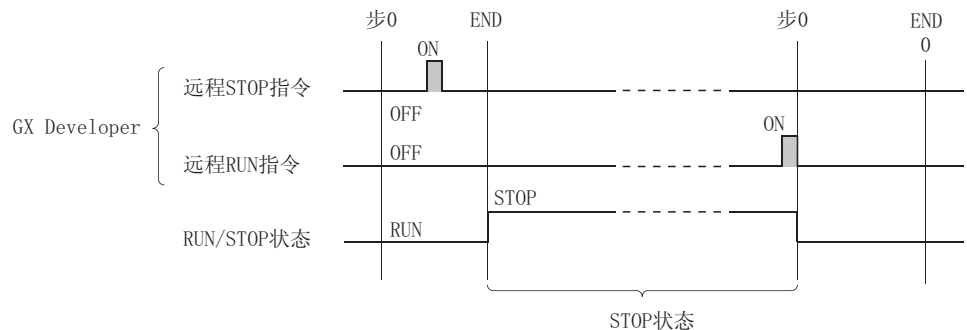


图 6.25 通过 GX Developer 进行的远程 RUN/STOP

(5) 注意事项

由于 CPU 模块是 STOP 优先，因此应注意以下几点。

(a) 变为 STOP 状态的时机

通过 RUN 触点或 GX Developer 中的任意一个进行远程 STOP 时，CPU 模块将变为 STOP 状态。

(b) 远程 STOP 后再次使其为 RUN 状态时

通过远程 STOP 使 CPU 模块进入 STOP 状态后，再次使其变为 RUN 状态时，应按最初进行的远程 STOP 的步骤进行远程 RUN。

☒ 要 点

1. RUN/STOP 状态如下所示。

- RUN 状态 对顺控程序的步 0 ~ END 指令的运算反复执行的状态。
- STOP 状态 是停止顺控程序运算的状态，输出 (Y) 全部为 OFF。

2. 如果进行了 CPU 模块的复位，复位后将变为 RUN/STOP/RESET 开关中设置的状态。

6.12.2 远程 RESET(远程复位)

(1) 关于远程 RESET

远程 RESET 是指, CPU 模块为 STOP 状态时, 通过 GX Developer 的操作对 CPU 模块进行复位的操作。

此外, 即使 RUN/STOP/RESET 开关位于 RUN 的位置, 由于发生了自诊断功能检测出的出错使 CPU 模块处于停止状态时, 可以进行复位。

(2) 远程 RESET 的用途

在 CPU 模块位于较远位置的状态下发生了出错时, 可以通过 GX Developer 对 CPU 模块进行远程 RESET。

(3) 远程 RESET 方法

只有通过 GX Developer 的操作才可以进行远程 RESET。

远程 RESET 的操作步骤如下所示。

- CPU 模块为 RUN 状态时, 通过远程 STOP 操作使其变为 STOP 状态。
- 通过远程 RESET 操作对 CPU 模块进行复位。

通过 GX Developer 的 [Online(在线)] → [Remote operation(远程操作)] 进行。

(4) 注意事项

(a) RUN 状态时的远程 RESET

CPU 模块为 RUN 状态时, 不能通过远程 RESET 进行复位。

应通过远程 STOP 操作使 CPU 模块变为 STOP 状态后再进行远程 RESET。

(b) 远程处理结束后的状态

在进行了远程 RESET 的 CPU 模块中复位处理结束后, CPU 模块将变为 RUN/STOP/RESET 开关中所设置的运行状态。

- RUN/STOP/RESET 开关位于 STOP 位置时, 复位结束后 CPU 模块将变为 STOP 状态。
- RUN/STOP/RESET 开关位于 RUN 位置时, 复位结束后 CPU 模块将变为 RUN 状态。

(c) 噪声导致发生了异常时

应注意由于噪声导致 CPU 模块发生了异常时, 有时会发生不能通过远程 RESET 进行复位的现象。

在不能通过远程 RESET 进行复位的情况下, 应通过 RUN/STOP/RESET 开关进行复位, 或者重新启动可编程控制器的电源。

☒ 要 点

1. 应注意进行了远程 RESET 时，复位处理结束后 CPU 模块将变为 RUN/STOP/RESET 开关中所设置的运行状态。
 2. 即使未在可编程控制器参数的可编程控制器系统设置中将远程复位设置为“允许”，GX Developer 的远程处理也会结束。
但是，由于在 CPU 中不能进行复位处理，因此不能复位。
通过 GX Developer 进行了远程 RESET 后 CPU 模块的状态也未发生变化时，应在可编程控制器参数的可编程控制器系统设置中确认远程复位是否处于“允许”状态。
-

6.12.3 远程操作与 CPU 模块的 RUN/STOP 状态的关系

(1) 远程操作与 CPU 模块的 RUN/STOP 状态的关系

根据远程操作与 CPU 模块的 RUN/STOP 状态的组合，CPU 模块的动作状态如表 6.18 所示。

表 6.18 RUN/STOP 状态与远程操作的关系

RUN/STOP 状态	远程操作		
	RUN *1	STOP	RESET
RUN	RUN	STOP	不能操作 *2
STOP	STOP	STOP	RESET *3

*1: 通过 RUN 触点进行操作时，需要在可编程控制器参数的可编程控制器系统设置中进行“RUN-PAUSE 触点的设置”。

*2: 将 CPU 模块通过远程 STOP 操作变为 STOP 状态的情况下，可以进行复位。

*3: CPU 模块由于出错而处于停止的状况下也包含在内。

(2) 来自于同一个 GX Developers 的远程操作

当通过同一个 GX Developers 进行了远程操作时，最后执行的远程操作的状态有效。

6.13 监视功能

(1) 关于监视功能

监视功能是指，通过 GX Developer 对 CPU 模块的程序、软元件、智能功能模块的状态进行读取的功能。

可执行的监视功能如下所示。

- 梯形图监视
- 软元件 / 缓冲存储器批量监视
- 软元件测试
- 程序一览监视
- 梯形图登录监视

关于 GX Developer 的监视功能，请参阅以下手册。

- GX Developer 操作手册

(2) 监视请求的处理时机及显示数据

CPU 模块通过 END 处理执行来自于 GX Developer 的监视请求的处理。

因此在 GX Developer 中，将显示 CPU 模块的 END 处理时的数据。

6.14 CPU 模块运行中的程序写入

在 CPU 模块中，可以进行梯形图模式下的运行中写入。

6.14.1 梯形图模式下的运行中写入

(1) 关于梯形图模式下的运行中写入

梯形图模式下的运行中写入是在 CPU 模块的运行过程中进行程序写入的功能。

梯形图模式下的运行中写入只能在测试模式时执行。

使用梯形图模式下的运行中写入，可以在不停止 CPU 模块程序运算的状况下更改程序。

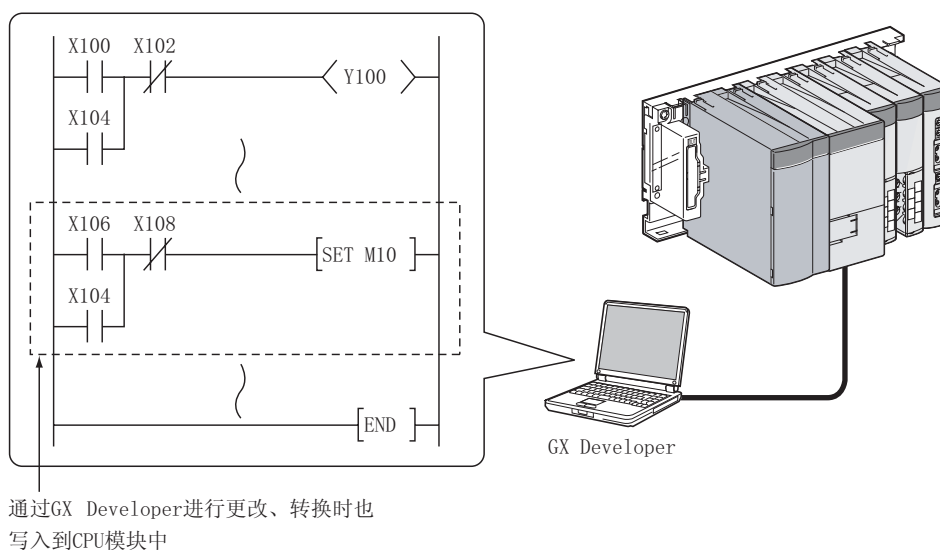


图 6.26 梯形图模式下的运行中写入的示意图

(2) 注意事项

运行中写入时的注意事项如下所示。

(a) 可进行运行中写入的存储器

可进行运行中写入的存储器只有程序存储器。

(b) 引导运行中的运行中写入

执行了运行中写入时，不能更改引导源的程序。

运行中写入后，进行可编程控制器的电源 OFF 或者 CPU 模块的复位之前，应将程序存储器的内容写入到内置 ROM 中。

(c) 一次运行中写入可写入的步数

一次运行中写入可写入的最多步数为 512 步。

(d) 运行中写入时的“运行中写入预留容量(步)”的更改

以下介绍更改运行中写入的“运行中写入预留容量(步)”时的注意事项。

1) 关于运行中写入预留容量(步)

在程序文件中，为了对应于程序文件容量有变化的运行中写入，包含有运行中写入预留容量(步)。

程序文件的容量为所创建的程序的容量加上“运行中写入预留容量(步)”的值。

2) 程序文件的容量大于预留的容量时

进行运行中写入时，程序文件的容量超出了预留容量(包含了运行中写入预留容量(步)的容量)的情况下，可以重新设置运行中写入时的“运行中写入预留容量(步)”。

因此，用户用存储区中有空余区域时，可以进行运行中写入。

3) 重新设置了运行中写入预留容量(步)时的扫描时间

应注意执行了运行中写入后扫描时间将延长。

关于扫描时间的延长时间，请参阅 10.1.3 节。

(e) 运行中写入时不能正常动作的指令

进行了运行中写入时，不能正常动作的指令有以下 2 种。

- 下降沿指令
- 上升沿指令

1) 下降沿指令

在运行中写入的写入范围内存在有下降沿指令的情况下，运行中写入结束时即使下降沿指令的执行条件 (ON → OFF) 未成立，也将执行下降沿指令。

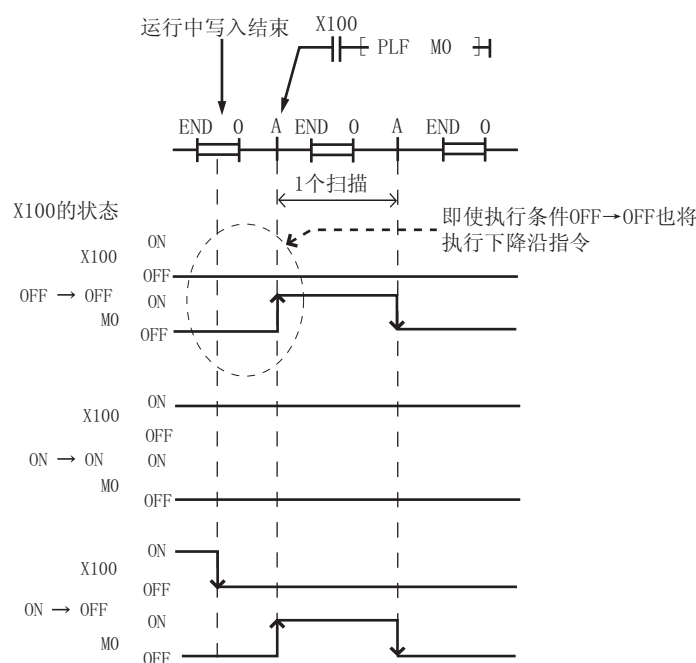


图 6.27 下降沿指令的动作

对象的下降沿指令为 LDF、ANDF、ORF、MEF、PLF 指令。

6.15 看门狗定时器 (WDT)

(1) 关于看门狗定时器 (WDT)

看门狗定时器是用于检测 CPU 模块的硬件及顺控程序异常的 CPU 模块内部的定时器。

(2) 看门狗定时器的设置及复位

(a) 看门狗定时器的设置

看门狗定时器的设置时间可以在可编程控制器参数的 PC RAS 设置中更改。

看门狗定时器的缺省值被设置为 200ms。

看门狗定时器可以在 10 ~ 2000ms(单位 :10ms) 的范围内更改。

(b) 看门狗定时器的复位

CPU 模块在 END 处理执行过程中对看门狗定时器进行复位。

- CPU 模块正常动作, 在看门狗定时器的设置值以内执行 END 指令时, 看门狗定时器不会发生“时间到”。
- 由于 CPU 模块的硬件异常导致顺控程序的扫描延长, 在看门狗定时器的设置值以内未能执行 END 指令时, 看门狗定时器将发生“时间到”。

(3) 看门狗定时器发生了“时间到”时

看门狗定时器“时间到”时将发生看门狗定时器出错, CPU 模块的动作情况如下所示。

- 1) CPU 模块的输出全部 OFF。
- 2) 前面的“RUN”LED 熄灯,“ERR.”LED 闪烁。
- 3) SM0、SM1 为 ON, SDO 中存储出错代码 5001(“WDT ERROR”)。

(4) 注意事项

(a) 关于看门狗定时器的误差

看门狗定时器的计量时间将产生 0 ~ 10ms 范围的误差。

设置看门狗定时器时，应设置考虑了上述误差的值。

☒ 要 点

1. 扫描时间是指，CPU 模块从顺控程序的步 0 开始执行运算起，至再次执行顺控程序的步 0 为止的时间。
每个扫描的扫描时间并不相同，扫描时间根据所使用的指令的执行与否而不一样。
2. 希望使每个扫描以相同的扫描时间执行时，应使用恒定扫描功能
( 6.9 节)。

6.16 通过 GX Developer 进行 CPU 模块的系统显示

将 GX Developer 与 CPU 模块相连接，通过系统监视可以确认以下项目。

- 安装状态
- 动作状态
- 模块详细信息
- 产品信息

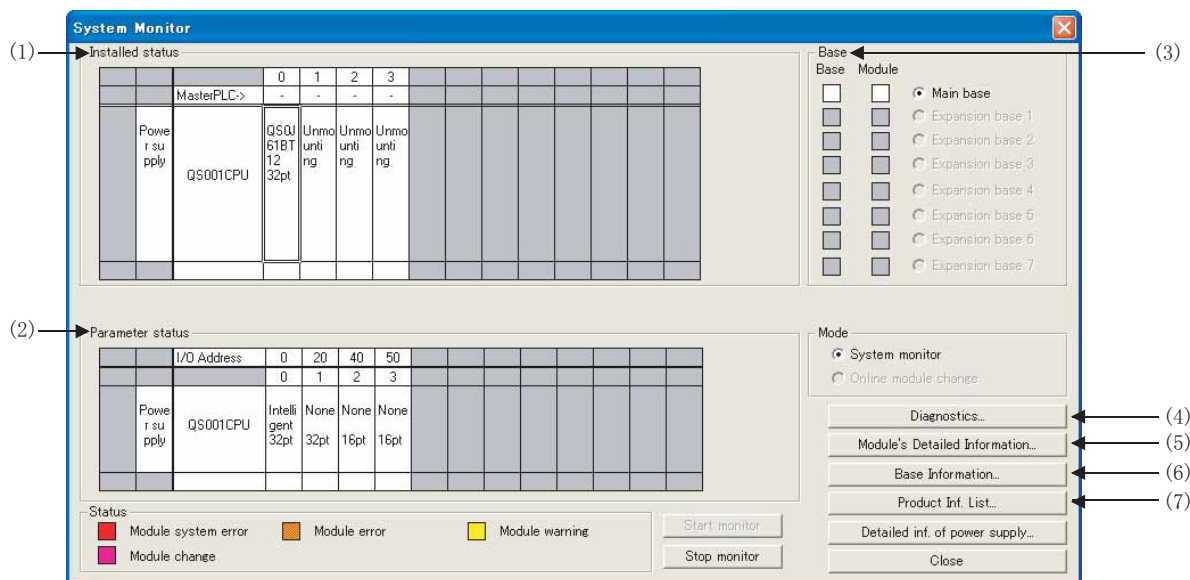


图 6.29 系统监视画面

(1) 安装状态

可以确认基板上安装的模块型号、点数。

未安装模块的插槽将显示为“未安装”。

对于在可编程控制器参数的 I/O 分配设置中被设置为“空闲”的插槽，即使安装了模块，也不显示模块型号。

(2) 参数状态

可以确认基板的各插槽的 I/O 地址号、模块类型、点数。

在参数状态中显示了空闲 0 点、分配出错时，表示可编程控制器参数的 I/O 分配与实际状态不符。

应使可编程控制器参数的 I/O 分配符合实际安装状态。

(3) 基板

可以确认基板及安装的模块的状态。

模块栏中即使只存在有 1 个异常的模块时，模块栏也将变为该异常模块的状态色。

(4) 诊断

用于确认所选择的模块的状态、出错情况。

6.17 LED 的显示

CPU 模块前面的 LED 显示 CPU 模块的动作状态。

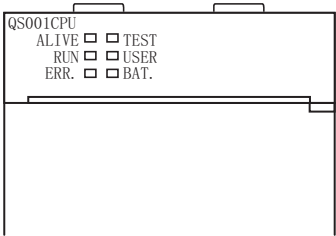


图 6.31 CPU 模块前面的 LED

备注

关于各 LED 的显示内容的详情，请参阅以下手册。
☞ QSCPU 用户手册（硬件设计 / 维护点检篇）

6.17.1 LED 的熄灯方法

对于亮灯的 LED 可以通过以下操作熄灯。（除复位操作以外。）

表 6.19 LED 的熄灯方法

熄灯方法	对象 LED			
	ERR.	USER	BAT.	BOOT
消除了出错原因后，操作特殊继电器 SM50、特殊寄存器 SD50 进行出错解除。 (限于继续运行出错) *1	○	○	○	×

○：有效 ×：无效

*1: 关于特殊继电器、特殊寄存器的内容
SM50: OFF → ON 时，进行 SD50 中存储的出错代码的出错解除。
SD50: 存储出错解除的出错代码。
 关于出错代码，请参阅以下手册。
☞ QSCPU 用户手册（硬件设计 / 维护点检篇）

备忘录

[illegible]

第 7 章 与智能功能模块的通信

7.1 与 CC-Link Safety 主站模块的通信

CPU 模块与 CC-Link Safety 主站模块的通信是通过自动刷新进行的。
进行自动刷新时，在 GX Developer 的网络参数的 CC-Link 设置中设置远程输入、远程输出、远程寄存器的刷新软元件。
关于网络参数的 CC-Link 设置的设置项目，请参阅 8.2 节。

备注

关于网络参数的 CC-Link 设置的详细内容，请参阅以下手册。

 CC-Link Safety 主站模块用户手册 (详细篇)

1

概要

2

性能规格

3

顺控程序的执行

4

I/O 地址号的分配

5

关于 CPU 模块中使用的存储器及文件

6

功能

7

与智能功能模块的通信

8

参数

7.2 与 MELSECNET/H 模块的通信

CPU 模块与 MELSECNET/H 模块的通信是通过自动刷新进行的。
进行自动刷新时，在 GX Developer 的网络参数的 MELSECNET/H 设置中设置刷新参数。
关于网络参数的 MELSECNET/H 设置的设置项目，请参阅 8.2 节。

☒ 要 点

在安全 CPU 模块中使用 MELSECNET/H 模块时，可使用的功能是有限制的。
关于使用 MELSECNET/H 模块时的限制，请参阅附录 4。

备注

关于网络参数的 MELSECNET/H 设置的详细内容，请参阅以下手册。
☞ Q 系列 MELSECNET/H 网络系统参考手册 (PLC 网络篇)

第 8 章 参数

本章介绍构筑可编程控制器系统时设置的参数有关内容。

(1) 参数的类型

CPU 模块的参数有以下 2 种。

- 可编程控制器参数 (☞ 8.1 节)
单独使用可编程控制器时设置。
- 网络参数 (☞ 8.2 节)
将 CC-Link Safety 主站模块、MELSECNET/H 模块与可编程控制器组合使用时设置。

(2) 参数的设置方法

参数是在 GX Developer 中设置。

关于 GX Developer 中的基本操作, 请参阅以下手册。

☞ GX Developer 操作手册

☒ 要 点

GX Developer 中灰色显示 (不能选择) 的设置项目是所使用的 CPU 模块中不存在的功能, 因此无需设置。

备 注

- 本章的表中显示参数号在参数设置中发生了出错时被存储到特殊寄存器 (SD16 ~ 26) 中。
应根据参数号确定参数出错的位置。
关于参数号的一览, 请参阅附录 3。
- 关于参数的存储步骤, 请参阅第 11 章。

8.1 可编程控制器参数

本节介绍可编程控制器参数的一览及各参数的内容。

(1) 可编程控制器名设置

设置所使用的 CPU 模块的标识、注释。

在可编程控制器名设置中即使设置了标识、注释也不会影响实际的动作。

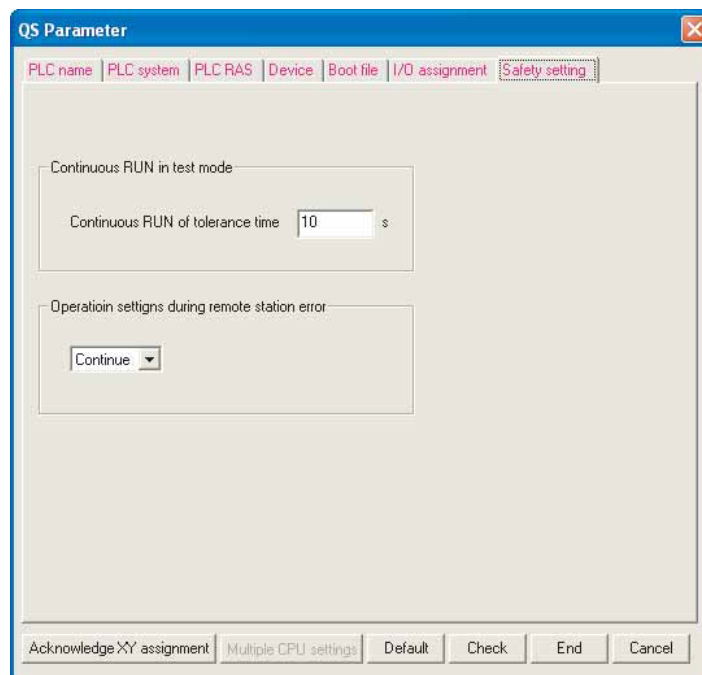


图 8.1 可编程控制器名设置

表 8.1 可编程控制器名设置一览

项目	参数号	内容	设置范围	缺省值	参阅章节
标识	0000 _H	设置 CPU 模块的标识 (名称、用途)。	最多 10 个半角字符	无设置	----
注释	0001 _H	设置 CPU 模块的标识的注释。	最多 64 个半角字符	无设置	----

(2) 可编程控制器系统设置

进行使用 CPU 模块时的必要设置。

在保持缺省值不变的情况下也可进行控制。

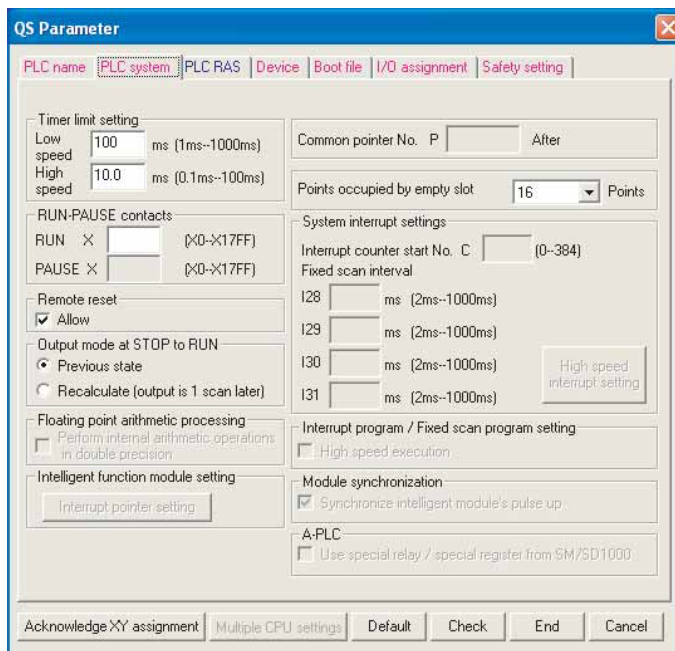


图 8.2 可编程控制器系统设置

表 8.2 可编程控制器系统设置一览

项目	参数号	内容	设置范围	缺省值	参阅章节
定时器时限 设置	1000 _H	设置低速定时器 / 高速定时器的 时限	1ms ~ 1000ms(单位 :1ms)	100ms	9.2.8 节
			0.1ms ~ 100.0ms(单位 :0.1ms)	10.0ms	9.2.8 节
RUN-PAUSE 触点	1001 _H	设置 CPU 模块的 RUN 控制触点。	X0 ~ 17FF	无设置	6.12.1 节
远程复位	1002 _H	设置通过 GX Developer 的远程复 位操作的允许 / 禁止。	允许 / 不允许	允许	6.12.2 节
STOP ~ RUN 时的输出模式	1003 _H	设置从 STOP 状态切换到 RUN 状态 时的输出 (Y) 状态。	输出 STOP 前的输出 (Y) 状态 / 输 出 (Y) 清除 (输出 1 个扫描后)	输出 STOP 前的输出 (Y) 状态	6.10 节
空插槽点数	1007 _H	设置主基板的空插槽点数。	0 点 /16 点 /32 点 /64 点 /128 点 /256 点 /512 点 /1024 点	16 点	4.2.1 节

(3) PLC RAS 设置

设置用于 RAS 功能的各种设置。

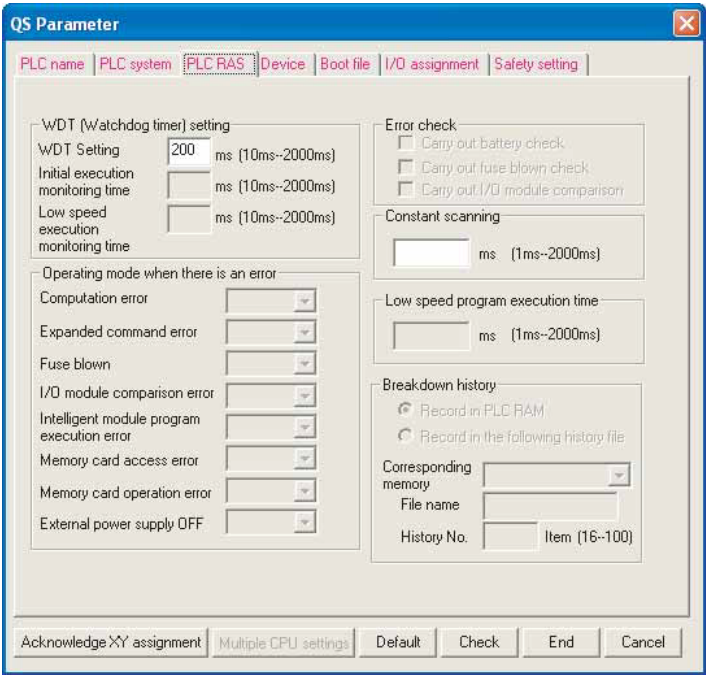


图 8.3 PLC RAS 设置

表 8.3 PLC RAS 设置一览

项目		参数号	内容	设置范围	缺省值	参阅章节
WDT (看门狗定时器) 设置	WDT 设置	3000 _H	设置 CPU 模块的看门狗定时器的时间。	10ms ~ 2000ms(单位 :10ms)	200ms	3.2 节
恒定扫描		3003 _H	设置恒定扫描时间。	1ms ~ 2000ms(单位 :10ms)	无设置	6.9 节

(4) 软元件设置

设置各软元件的使用点数。

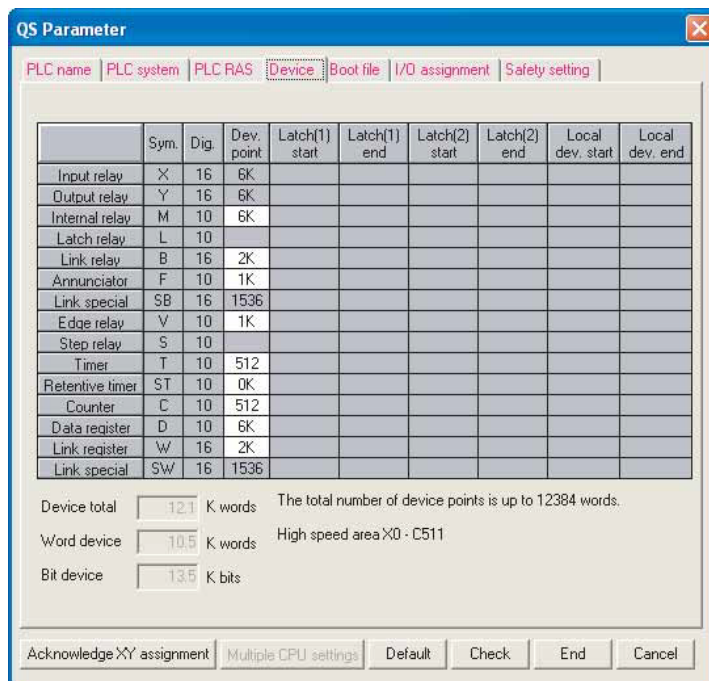


图 8.4 软元件设置

表 8.4 软元件设置一览

项目	参数号	内容	设置范围	缺省值	参阅章节
软元件点数	2000 _H	根据系统设置软元件的使用点数。	固定为 X(6k 点)、Y(6k 点)、SB(1536 点)、SW(1536 点)。 可以在包含上述点数 (2400 字) 在内的合计 12384 字的范围内设置。 • 1 软元件：最多 32k 点	X : 6k 点 Y : 6k 点 M : 6k 点 B : 2k 点 F : 1k 点 SB : 1536 点 V : 1k 点 T : 512 点 ST : 0k 点 C : 512 点 D : 6k 点 W : 2k 点 SW : 1536 点	9.1 节

(5) 引导文件设置
 设置是否进行来自于内置 ROM 的引导。

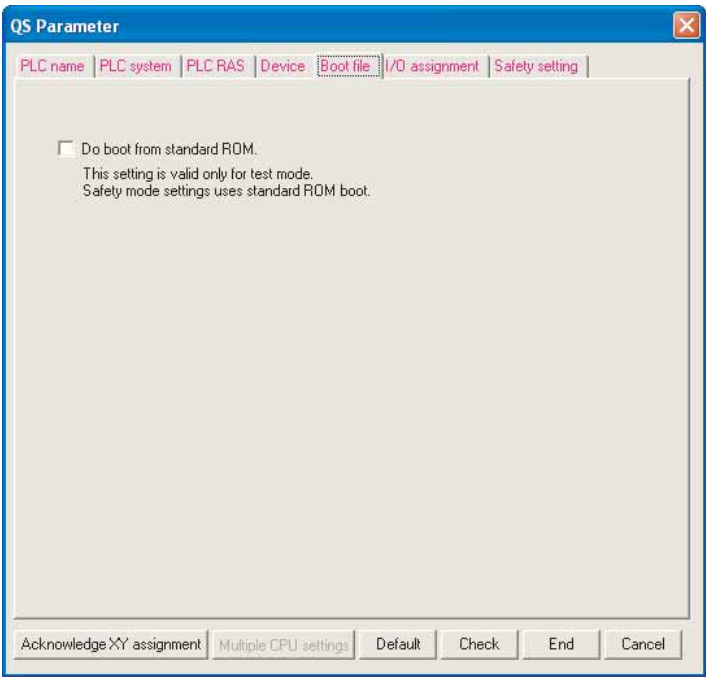


图 8.5 引导文件设置

表 8.5 引导文件设置一览

项目	参数号	内容	设置范围	缺省值	参阅章节
引导文件设置	7000 _H	设置测试模式时，是否进行来自于内置 ROM 的引导。	进行引导 / 不进行引导	不进行引导	5.1.4 节

☒ 要 点

安全模式时，与引导文件设置无关，进行引导运行。

(6) I/O 分配设置

设置系统的各模块的安装状态。

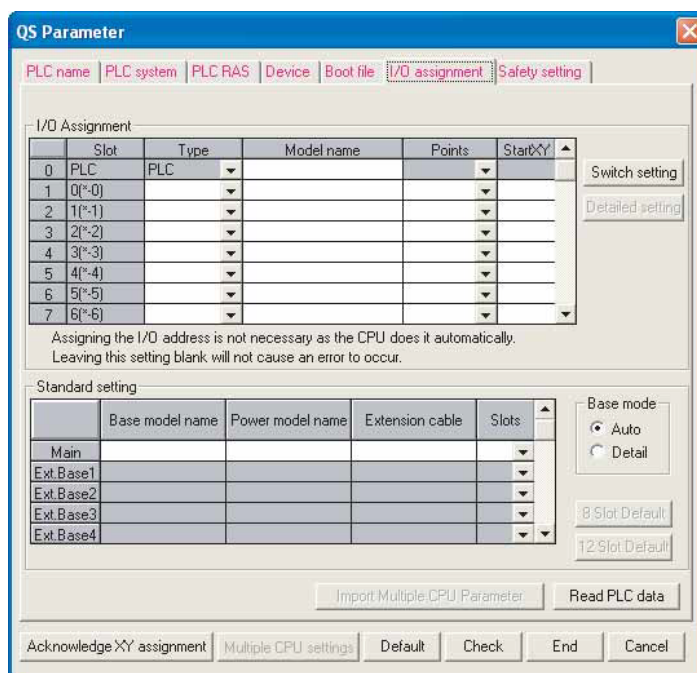


图 8.6 I/O 分配设置

表 8.6 I/O 分配设置一览

项目	参数号	内容	设置范围	缺省值	参阅章节
I/O 分配	类型	设置安装的模块的类型。	空闲 / 智能	无设置	4.3 节
	型号	设置安装的模块的型号。(作为用户的备忘。CPU 模块不使用。)	半角 16 个字符	无设置	
	点数	设置各插槽的点数。	0 点 / 16 点 / 32 点 / 48 点 / 64 点 / 128 点 / 256 点 / 512 点 / 1024 点	无设置	
	起始 XY (起始 I/O 地址号)	设置各插槽的起始 I/O 地址号。	0 _H ~ 3F0 _H	无设置	
基本设置	基板型号	设置所使用的主基板的型号。 (作为用户的备忘。CPU 模块不使用。)	半角 16 个字符	无设置	4.4 节
	电源模块 型号	设置主基板上安装的电源模块的 型号。(作为用户的备忘。CPU 模 块不使用。)	半角 16 个字符	无设置	
	扩展电缆 型号	设置扩展电缆的型号。(作为用 户的备忘。CPU 模块不使用。)	半角 16 个字符	无设置	
	插槽数	设置主基板的插槽数。	4	无设置	
开关设置	0407 _H	禁止使用	禁止使用	----	----

(7) X/Y 分配确认

在 I/O 分配、MELSECNET 设置、CC-Link 设置中确认设置内容。

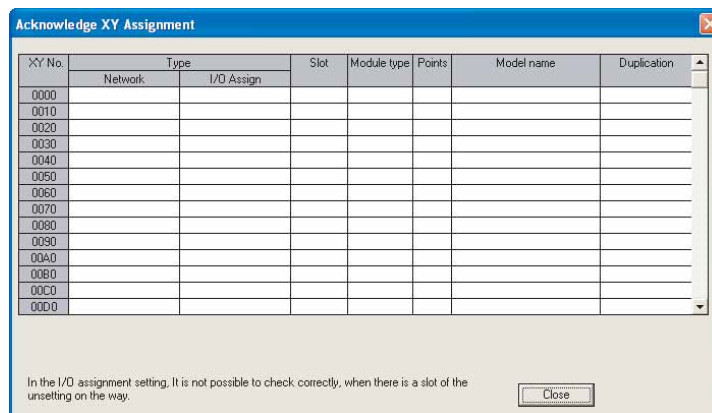


图 8.7 X/Y 分配确认

表 8.7 X/Y 分配确认一览

项目	参数号	内容	设置范围	缺省值	参阅章节
X/Y 分配确认	----	在 I/O 分配、MELSECNET 设置、CC-Link 设置中可以确认设置内容。	----	----	----

(8) 安全设置

进行测试模式下的连续运行、远程站异常时的动作设置。

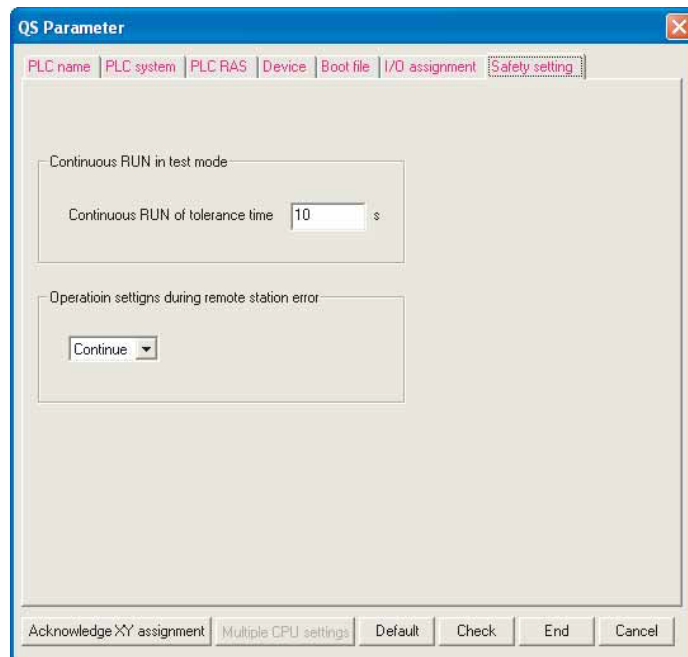


图 8.8 安全设置

表 8.8 安全设置

项目	参数号	内容	设置范围	缺省值	参阅章节
测试模式下的连续运行	6000 _H	设置测试模式下的连续运行允许时间。	1 秒～ 86400 秒	10 秒	6.5 节
远程站异常时的动作设置		设置远程站异常时的动作设置	停止 / 继续运行	停止	----

8.2 网络参数

本节介绍网络参数的一览及各参数的内容。

- 关于本节的参数号栏中所示的 mn、M、N
本节的参数号栏中所示的 mn、M、N 如下所示。

mn : 表示起始 I/O 地址号 $\div 16$ 的值。
N : 表示是第几个模块。
M : 表示网络的类型。

表 8.9 MELSECNET/H 设置时 ( 本节 (1))

M	网络类型
2 _H	MELSECNET/10 模式 (常规站)、MELSECNET/H 模式 (常规站)、MELSECNET/H 扩展模式 (常规站)

表 8.10 CC-Link 设置时 ( 本节 (2))

M	网络类型
0 _H	主站

(1) MELSECNET/H 设置
设置 MELSECNET/H 的网络参数。

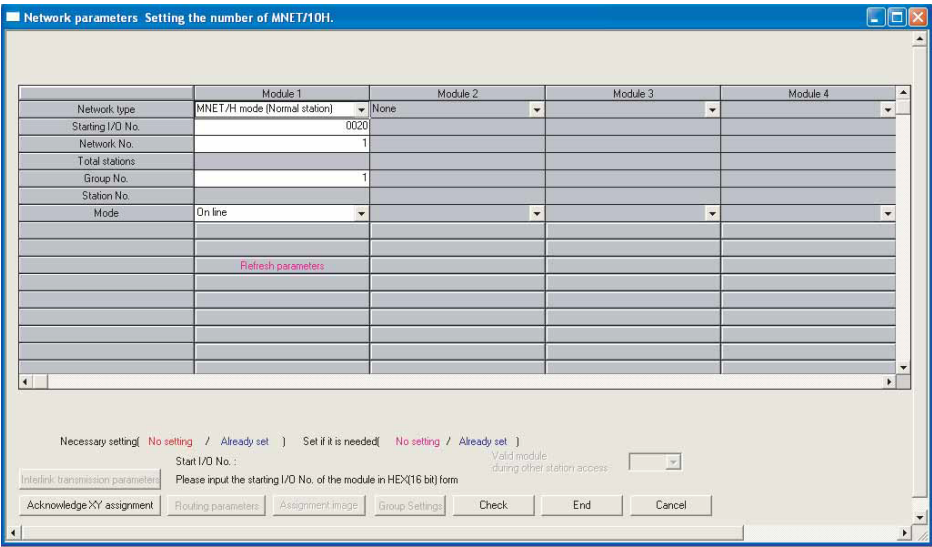


图 8.9 MELSECNET/10H 个数设置

表 8.11 MELSECNET/H 设置一览

项目	参数号	内容	设置范围	缺省值	参阅章节
MELSECNET 个数设置	5000 _H	设置 MELSECNET/H 的网络参数。	参阅 Q 系列 MELSECNET/H 的手册	----	----
起始 I/O 地址号	5NM0 _H				
网络号					
组号					
模式					
刷新参数					
	05mn _H				
	5NM0 _H				
	5NM1 _H				

(2) CC-Link 设置

设置 CC-Link 的参数。

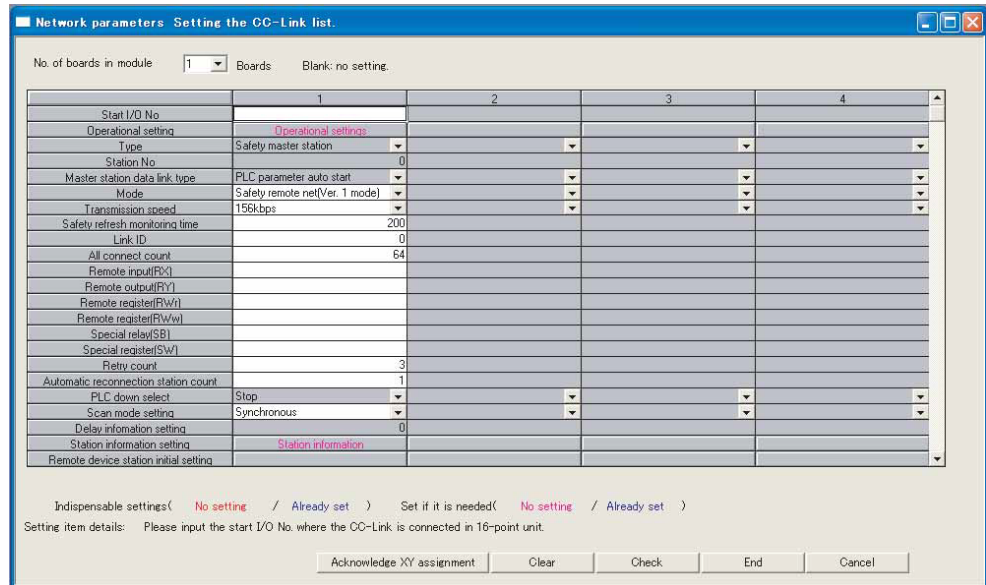


图 8.10 CC-Link 设置

表 8.12 CC-Link 设置一览

项目	参数号	内容	设置范围	缺省值	参阅章节
模块个数设置	C000 _H	设置 CC-Link Safety 的参数	参阅 CC-Link Safety 的手册	----	----
起始 I/O 地址号	CNM2 _H				
动作设置					
模式设置					
传输设置					
安全刷新监视时间					
链接 ID					
总链接个数					
远程输入 (RX) 刷新软元件	CNM1 _H				
远程输出 (RY) 刷新软元件					
远程寄存器 (RW _r) 刷新软元件					
远程寄存器 (RW _w) 刷新软元件					
特殊继电器 (SB) 刷新软元件					
特殊寄存器 (SW) 刷新软元件					
重试次数	CNM2 _H				
自动复原个数					
扫描模式指定					
站信息设置					

第 9 章 软元件说明

本章介绍 CPU 模块中可使用的软元件。

9.1 软元件一览

CPU 模块中可使用的软元件名及使用范围如表 9.1 所示。

表 9.1 软元件一览表

分类	类型	软元件名	缺省值		参数设置的 设置范围	参阅章节
			点数	使用范围		
内部用户 软元件	位 软元件	输入	6144 点	X0 ~ 17FF	可以在 12384 字 以内更改 *2	9.2.1 节
		输出	6144 点	Y0 ~ 17FF		9.2.2 节
		内部继电器	6144 点	M0 ~ 6143		9.2.3 节
		报警器	1024 点	F0 ~ 1023		9.2.4 节
		变址继电器	1024 点	V0 ~ 1023		9.2.5 节
		链接继电器	2048 点	B0 ~ 7FF		9.2.6 节
		链接特殊继电器	1536 点	SB0 ~ 5FF		9.2.7 节
	字 软元件	定时器 *1	512 点	T0 ~ 511		9.2.8 节
		累计定时器 *1	0 点	--		
		计数器 *1	512 点	C0 ~ 511		9.2.9 节
		数据寄存器	6144 点	D0 ~ 6143		9.2.10 节
		链接寄存器	2048 点	W0 ~ 7FF		9.2.11 节
		链接特殊寄存器	1536 点	SW0 ~ 5FF		9.2.12 节
内部系统 软元件	位 软元件	特殊继电器	5120 点	SM0 ~ 5119	禁止	9.3.1 节
	字 软元件	特殊寄存器	5120 点	SD0 ~ 5119		9.3.2 节
嵌套	--	嵌套	15 点	N0 ~ 14	禁止	9.4 节
常数	--	10 进制数	K-2147483648 ~ 2147483647			9.5.1 节
		16 进制数	H0 ~ FFFFFFFF			9.5.2 节

*1 : 对于定时器、累计定时器、计数器, 触点 / 线圈为位软元件, 当前值为字软元件。

*2 : 在 GX Developer 的可编程控制器参数中可以更改。(输入、输出、链接特殊继电器、链接特殊寄存器除外。 (9.2 节)

9.2 内部用户软元件

(1) 关于内部用户软元件

内部用户软元件是根据用户的用途可使用的软元件。
内部用户软元件事先设置了可使用的点数（缺省值）。
通过可编程控制器参数的软元件设置可以更改使用点数。

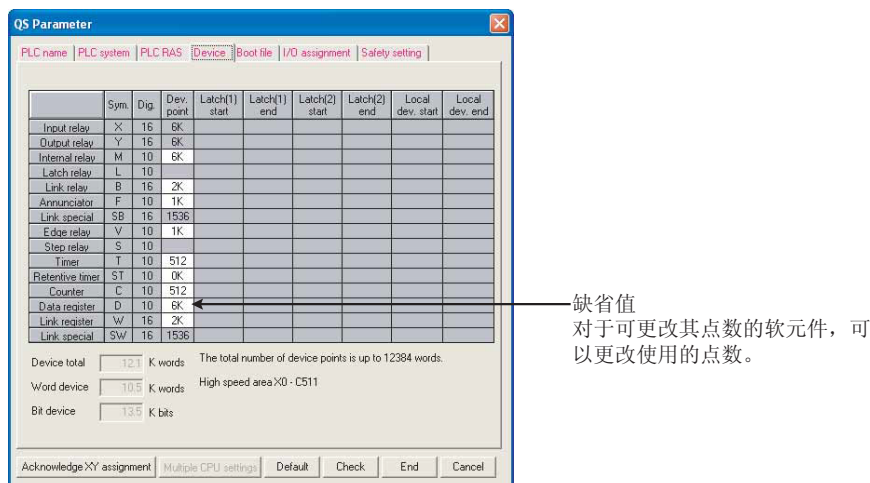


图 9.1 可编程控制器参数的软元件设置

(2) 内部用户软元件的设置范围

对于除 CPU 模块的输入 (X)、输出 (Y)、链接特殊继电器 (SB)、链接特殊寄存器 (SW) 以外的内部用户软元件，可以通过可编程控制器参数的软元件设置在 9.75k 字的范围更改使用点数。
更改内部用户软元件点数时的思路如下所示。

(a) 关于设置范围

- 1 个软元件是以 16 点为单位进行设置。
- 1 个软元件的最大点数为 32k 点。
- 定时器、累计定时器、计数器的 1 点是按线圈 1 点及触点 1 点的 2 点计算。

(3) 存储器容量的思路

设置内部用户软元件时，应满足以下公式。

$$(\text{位软元件容量}) + (\text{字软元件容量}) + (\text{定时器、累计定时器、计数器容量}) \leq 12384 \text{ 字}$$

(a) 位软元件时

对于位软元件，将 16 点作为 1 个字计算。

$$(\text{位软元件容量}) = \frac{(\text{X+Y+M+B+F+SB+V 的合计点数})}{16} \text{ (字)}$$

(b) 定时器 (T)、累计定时器 (ST)、计数器 (C) 时

对于定时器 (T)、累计定时器 (ST)、计数器 (C)，将 16 点作为 18 个字计算。

$$(\text{定时器、累计定时器、计数器容量}) = \frac{(\text{T、ST、C 的合计点数})}{16} \times 18 \text{ (字)}$$

(c) 字软元件时

对于数据寄存器 (D)、链接寄存器 (W)，是将 16 点作为 16 个字计算。

$$(\text{字软元件容量}) = \frac{(\text{D、W、SW 的合计点数})}{16} \times 16 \text{ (字)}$$

☒ 要 点

在参数中更改了内部用户软元件的使用点数时，对由更改前的参数所创建的顺控程序不能原封不动地使用。

在更改了内部用户软元件的使用点数的情况下，应将参数及顺控程序写入到 CPU 模块中。

(4) 软元件点数分配示例

软元件点数分配示例如表 9.2 所示。

表 9.2 软元件点数分配示例

软元件名	符号	进制数	软元件点数 *1*2		限制检查		
			点数	编号	容量 (字) *3		点数位 *2
输入继电器	X	16	6k(6144) 点	X0000 ~ 17FF	÷ 16	384 字 × 1	6144 点
输出继电器	Y	16	6k(6144) 点	Y0000 ~ 17FF	÷ 16	384 字 × 1	6144 点
内部继电器	M	10	8k(8192) 点	M0 ~ 8191	÷ 16	512 字 × 1	8192 点
链接继电器	B	16	1k(1024) 点	B0000 ~ 03FF	÷ 16	64 字 × 1	1024 点
报警器	F	10	1k(1024) 点	F0 ~ 1023	÷ 16	64 字 × 1	1024 点
链接特殊继电器	SB	16	1.5k(1536) 点	SB0000 ~ 05FF	÷ 16	96 字 × 1	1536 点
变址继电器	V	10	1k(1024) 点	V0 ~ 1023	÷ 16	64 字 × 1	1024 点
定时器	T	10	1k(1024) 点	T0 ~ 1023	$\times \frac{18}{16}$	1152 字 × 2	2048 点
累计定时器	ST	10	1k(1024) 点	ST0 ~ 1023	$\times \frac{18}{16}$	1152 字 × 2	2048 点
计数器	C	10	1k(1024) 点	C0 ~ 1023	$\times \frac{18}{16}$	1152 字 × 2	2048 点
数据寄存器	D	10	4k(4096) 点	D0 ~ 4095	× 1	4096 字	--
链接寄存器	W	16	1k(1024) 点	W0000 ~ 03FF	× 1	1024 字	--
链接特殊寄存器	SW	16	1.5k(1536) 点	SW0000 ~ 05FF	× 1	1536 字	--
软元件合计					11680 字 (12384 字以内)		31232 点

*1 : ☐ 的点数为系统固定。(不能更改)

*2 : 1 个软元件的最大点数为 32k 点。

*3 : 记入软元件点数乘以 (或者除以) 容量 (字) 栏中显示的数字后的数值。

9.2.1 输入 (X)

(1) 关于输入

输入是指，通过切换开关、限位开关、数字开关等的外部设备将指令及数据赋予到 CPU 模块中。

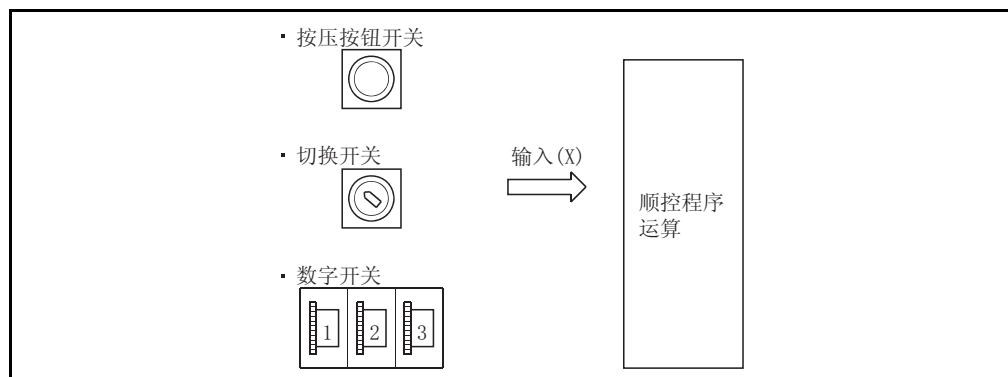


图 9.2 通过外部设备输入到 CPU 模块中的指令

(2) 输入 (X) 的思路

将 CPU 模块内置虚拟继电器 Xn 设想为输入点，在程序中使用该 Xn 的常开触点、常闭触点。

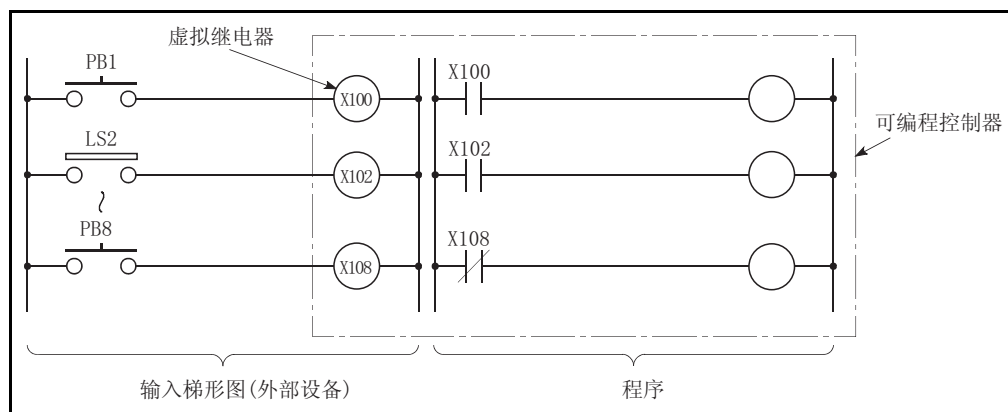


图 9.3 输入 (X) 的思路

(3) 常开触点与常闭触点的使用数

在程序内的 Xn 的常开触点与常闭触点的使用数在程序容量的范围内无限制。

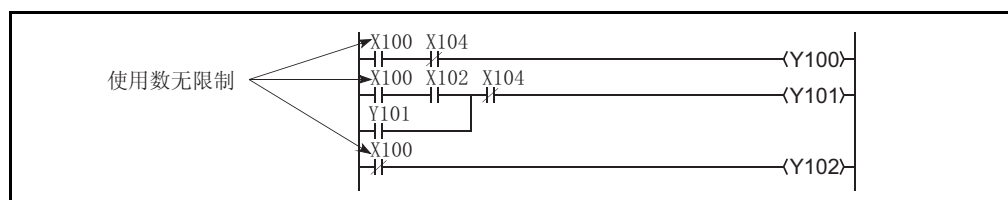


图 9.4 输入 (X) 在程序的使用

☒ 要 点

对所创建的程序进行调试时，可以通过以下方法对输入 (X) 进行 ON/OFF。

- GX Developer 的软元件测试
- OUT Xn 指令

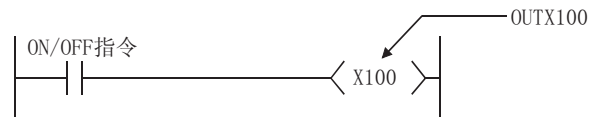


图 9.5 通过 OUT Xn 指令进行的输入 (X) 的 ON/OFF

9.2.2 输出 (Y)

(1) 关于输出

输出是指，将程序的控制结果输出到外部的信号灯、数字显示器、电磁开闭器（接触器）、螺线管等。

输出相当于至外部的 1 个常开触点，可以将其取出。

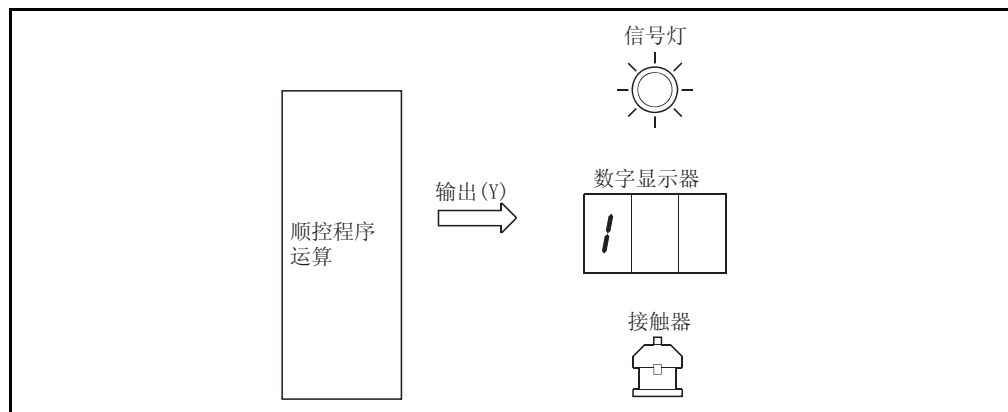


图 9.6 从 CPU 模块至外部设备的输出

(2) 常开触点与常闭触点的使用数

程序内的输出 Yn 的常开触点与常闭触点的使用数在程序容量的范围内无限制。

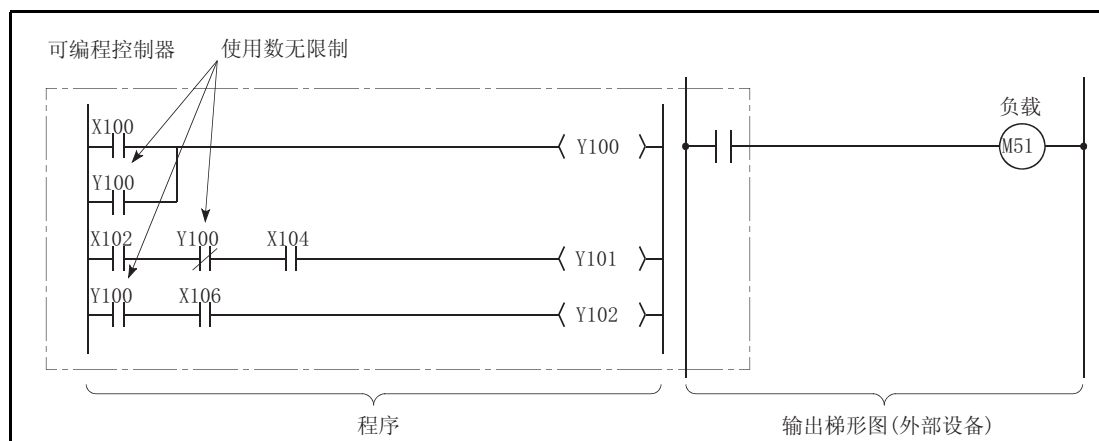


图 9.7 输出 (Y) 在程序中的使用

(3) 作为内部继电器 (M) 的替代品使用

可以将对应于未安装模块区域的输出 (Y) 作为内部继电器 (M) 的替代品使用。

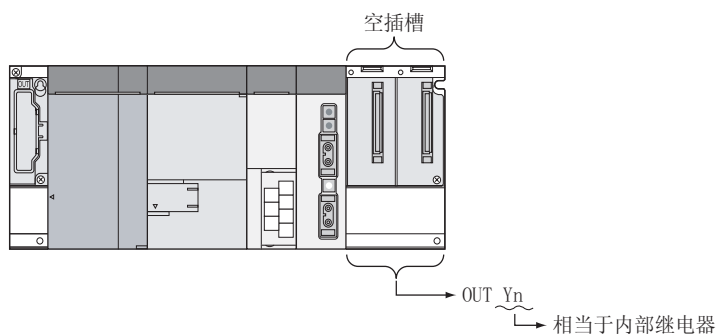


图 9.8 内部继电器的替代品

9.2.3 内部继电器 (M)

(1) 关于内部继电器 (M)

内部继电器是 CPU 模块内部使用的辅助继电器。

如果执行以下操作，内部继电器将全部 OFF。

- 可编程控制器的电源 OFF → ON 时
- CPU 模块的复位操作时

(2) 常开触点与常闭触点的使用数

程序内的触点（常开触点、常闭触点）的使用数在程序容量的范围内无限制。

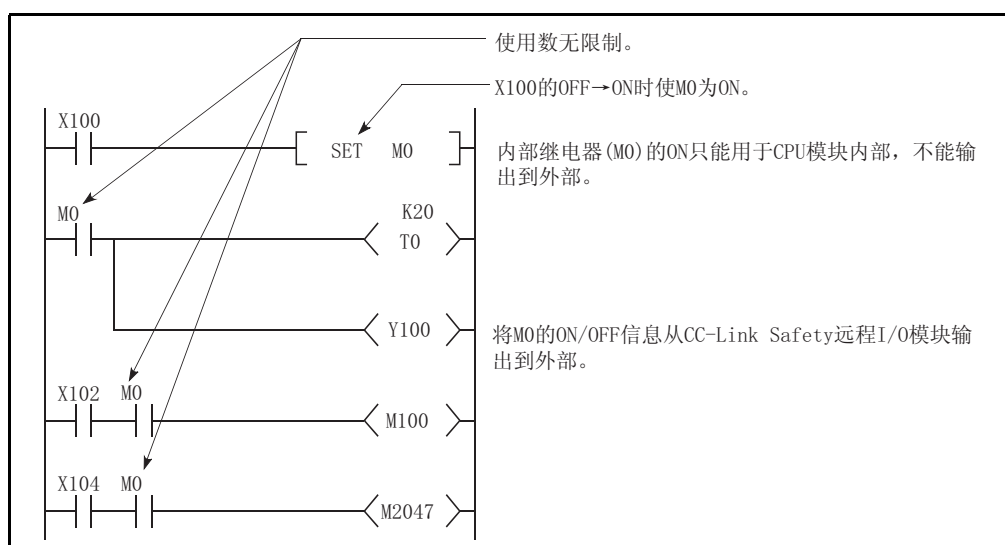


图 9.9 内部继电器在程序中的使用

(3) 输出至外部的的方法

将顺控程序的运算结果输出到外部时，使用输出 (Y)。

9.2.4 报警器 (F)

(1) 关于报警器

报警器是指，在用户创建的设备的异常 / 故障检测程序中使用的内部继电器。

(2) 报警器 ON 时的特殊继电器与特殊寄存器

报警器为 ON 时，特殊继电器 (SM62) 将 ON，特殊寄存器 (SD62 ~ 79) 中将存储变为 ON 的报警器的个数及编号。

- 特殊继电器 : SM62 即使只有 1 个报警器为 ON 时将 ON。
- 特殊寄存器 : SD62 存储最先为 ON 的报警器编号。
- SD63 存储变为 ON 状态的报警器的个数。
- SD64 ~ 79 以变为 ON 的顺序存储报警器的编号。
- (SD62 与 SD64 存储相同的报警器编号。)

此外，SD62 中存储的报警器编号也将被登录到操作 / 故障历史记录存储区中。

☒ 要 点

在可编程控制器的电源为 ON 的状态下即使多个报警器为 ON，也只有 1 个报警器编号被存储到操作 / 故障历史记录存储区中。

如果在 CPU 模块中进行出错解除，可以将处于 ON 状态的其它报警编号存储到操作 / 故障历史记录存储区中。

(3) 报警器的用途

如果在故障检测程序中使用报警器，特殊继电器 (SM62) 为 ON 时，通过监视特殊寄存器 (SD62 ~ 79)，可以确认设备有无异常、故障 (报警器编号)。

例

报警器 (F5) 为 ON 时，将变为 ON 的报警器编号输出到外部的程序

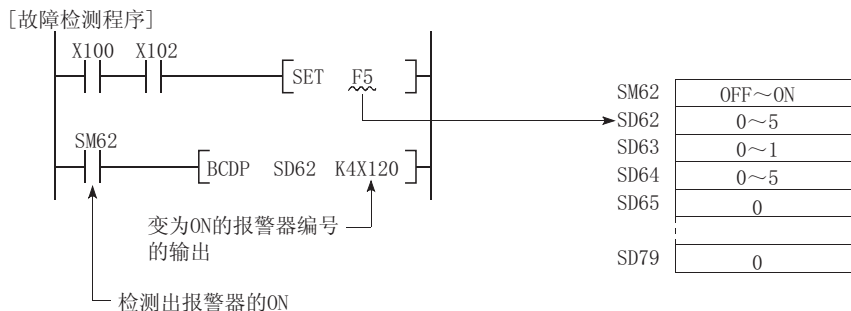


图 9.10 报警器的 ON 检测及存储

(4) 常开触点与常闭触点的使用数

程序内的触点 (常开触点、常闭触点) 的使用数在程序容量的范围内无限制。

(5) 使报警器为 ON 的方法及处理内容

(a) 使报警器为 ON 的方法

通过执行以下指令可以使报警器为 ON。

1) SET F $\square$ 指令

SET F $\square$ 指令只有在输入条件的上升沿时 (OFF → ON) 使报警器为 ON。

即使输入条件为 OFF，也保持报警器的 ON 状态不变。

经常使用报警器的情况下，通过使用 OUT F $\square$ 指令可以缩短扫描时间。

2) OUT F $\square$ 指令

通过使用 OUT F $\square$ 指令可以进行报警器的 ON/OFF，但由于每次执行扫描处理，因此比使用 SET F $\square$ 指令要慢。

此外，通过 OUT F $\square$ 指令使报警器为 OFF 时，需要执行 RST F $\square$ 指令，因此使报警器为 ON 时应使用 SET F $\square$ 指令。

☒ 要 点

通过除 SET F $\square$ 指令或者 OUT F $\square$ 指令以外 (例如 MOV 指令) 的指令使报警器为 ON 时，与内部继电器的动作相同。

不执行 SM62 的 ON 以及至 SD62、SD64 ~ 79 的报警器编号的存储。

(b) 报警器 ON 时的处理内容

1) 特殊寄存器 (SD62 ~ 79) 的存储数据

- 变为 ON 的报警器编号按 SD64 ~ 79 的顺序被存储。
- SD64 中存储的报警器编号被存储至 SD62 中。
- SD63 的内容被 +1。

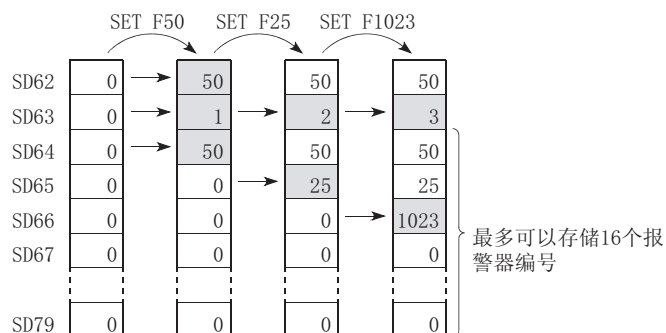


图 9.11 报警器 ON 时的处理内容

2) 设备本身的处理

模块前面的“USER”LED 将亮灯。

(6) 报警器的 OFF 方法及处理内容

(a) 报警器的 OFF 方法

通过执行以下指令可以使报警器为 OFF。

1) RST F□指令

RST F□指令仅在输入条件的上升沿 (OFF → ON) 时使报警器为 OFF。

通过 RST F□指令使报警器为 OFF 时，进行本项 (6) (b) 的报警器 OFF 时的处理。

2) OUT F□指令

虽然使用 OUT F□指令可以进行报警器的 ON/OFF，但由于执行各扫描处理，因此比使用 RST F□指令时要慢。

此外，即使通过 OUT F□指令使报警器编号 OFF，也不能执行本节的 (6) (b) 的报警器 OFF 时的处理。

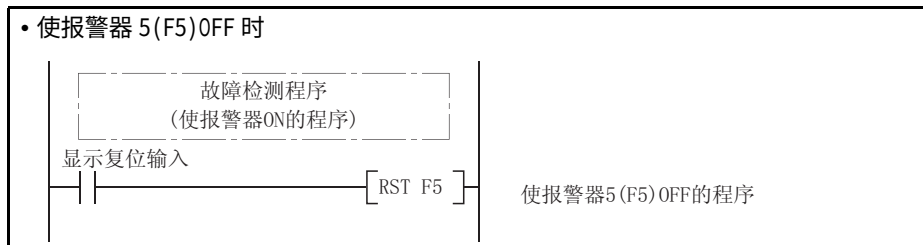


图 9.12 使报警器 OFF 时的示例

备注

关于各指令的详细内容，请参阅以下手册。

👉 QSCPU 编程手册 (公共指令篇)

(b) 报警器 OFF 时的处理内容

1) 通过 RST F 指令使报警器 OFF 时的特殊寄存器 (SD62 ~ 79) 的存储数据

- 通过 RST 指令删除指定的报警器编号，删除的报警器后面的报警器编号向前对齐。
- SD64中存储的报警器编号 OFF 时，新存储到 SD64 中的报警器编号将被存储到 SD62 中。
- SD63 的内容 -1。
- SD63 变为 0 时，SM62 将 OFF。

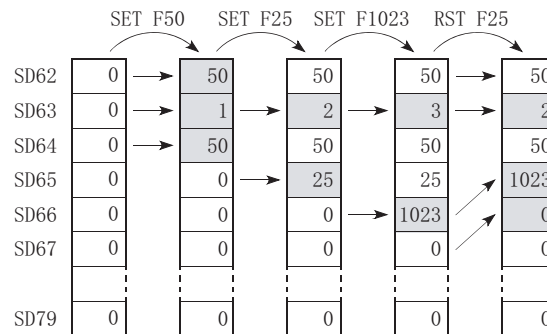


图 9.13 报警器 OFF 时的处理内容 (执行 RST F 指令时)

2) LED 的显示

SD64 ~ 79 的报警器编号全部为 OFF 时，由于报警器的 ON 而亮灯的 “ USER ” LED 将熄灯。

9.2.5 变址继电器 (V)

(1) 关于变址继电器

变址继电器是记忆从梯形图块的起始开始的运算结果 (ON/OFF 信息) 的软元件, 只能作为触点使用。

变址继电器不能被作为线圈使用。

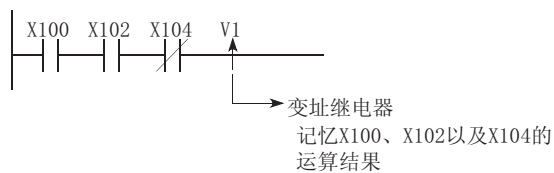


图 9.14 变址继电器

(2) 注意事项

在程序中, 不能使用多个相同编号的变址继电器。

9.2.6 链接继电器 (B)

(1) 关于链接继电器

链接继电器是指，将 MELSECNET/H 模块等的链接继电器 (LB) 刷新到 CPU 模块中时，或者将 CPU 模块内的数据刷新到 MELSECNET/H 模块等的链接继电器 (LB) 中时使用的 CPU 模块侧的继电器。

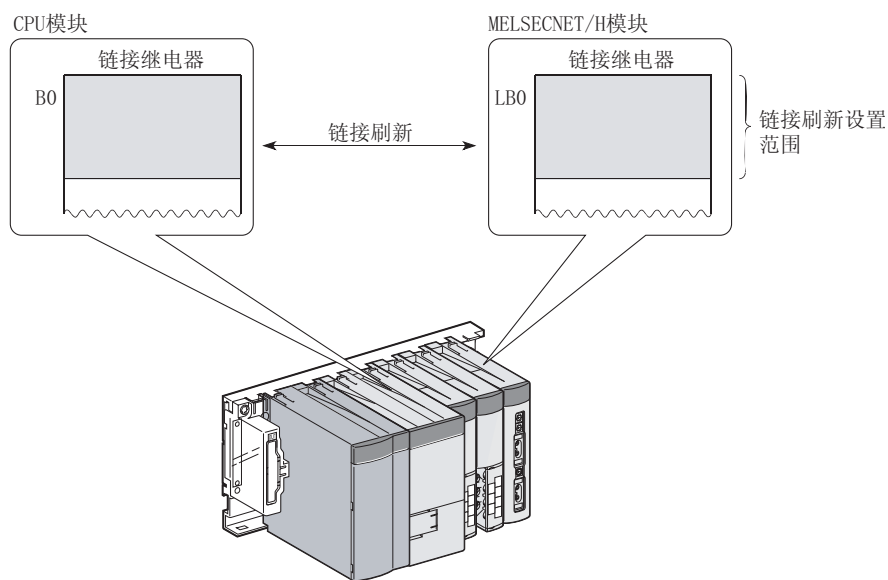


图 9.15 链接刷新

(2) 常开触点与常闭触点的使用数

在程序内的触点（常开触点、常闭触点）的使用数在程序容量的范围内无限制。

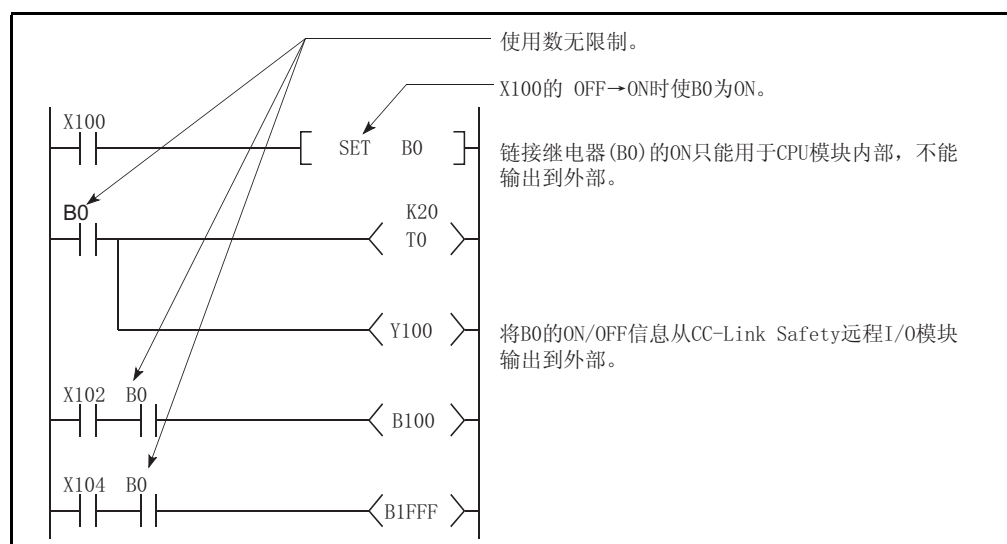


图 9.16 链接继电器

(3) 在网络系统中使用时

在网络系统中使用时，需要进行网络参数的设置。

对于未设置网络参数（MELSECNET/H 等网络系统中未使用）的范围，可以作为内部继电器使用。

☒ 要 点

MELSECNET/H 模块内的链接继电器为 16384 点，但 CPU 模块的链接继电器的缺省值为 2048 点。

使用上述所示的点数以后的链接继电器时，应在可编程控制器参数的软元件设置中更改链接继电器的点数。

备注

关于网络参数，请参阅以下手册。

☞ Q 系列 MELSECNET/H 网络系统参考手册（PLC 网络篇）

9.2.7 链接特殊继电器 (SB)

(1) 关于链接特殊继电器

链接特殊继电器是指，表示 CC-Link Safety 主站模块以及 MELSECNET/H 模块的通信状态、异常检测的继电器。

根据数据链接时发生的各种各样原因进行链接特殊继电器的 ON/OFF 控制。

通过监视链接特殊继电器，可以掌握数据链接的通信状态、异常状态等。

(2) 链接特殊继电器点数

链接特殊继电器的点数如表 9.3 所示。

表 9.3 各 CPU 模块的链接特殊继电器的点数

CPU 模块	链接特殊继电器点数
安全 CPU	1536 点 (SB0 ~ 5FF) . CC-Link Safety 主站模块以及 MELSECNET/H 模块的链接特殊继电器为 512 点。 链接特殊继电器可以按下图所示进行分配。 SB0 ⋮ SB1FF SB200 ⋮ SB3FF SB400 ⋮ SB5FF 链接特殊继电器 第1个网络模块用 第2个网络模块用 第3个网络模块用 512 点 512 点 512 点 } 1536 点

备注

关于链接特殊继电器的详细内容，请参阅以下手册。

☞ CC-Link Safety 主站模块用户手册 (详细篇)

☞ Q 系列 MELSECNET/H 网络系统参考手册 (PLC 网络篇)

9.2.8 定时器 (T)

(1) 关于定时器

定时器 (T) 是指，定时器的线圈为 ON 时开始计量，当前值达到设置值以上时发生“时间到”并使触点 ON 的软元件。

定时器为加法运算型。

定时器发生“时间到”时，当前值与设置值变为相同的值。

(2) 定时器的类型

定时器有线圈为 OFF 时当前值变为 0 的定时器，以及即使线圈为 OFF 时也保持当前值的累计定时器。

定时器中有低速定时器及高速定时器，累计定时器中有低速累计定时器及高速累计定时器。

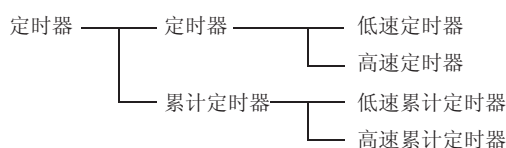


图 9.17 定时器的类型

(3) 定时器的使用方法

低速定时器与高速定时器为相同的软元件，通过定时器的指定（指令的格式）成为低速定时器或者高速定时器。例如，指定 OUT T0 时成为低速定时器，指定 OUTH T0 时成为高速定时器。

低速累计定时器与高速累计定时器为相同的软元件，通过定时器的指定（指令的格式）成为低速累计定时器或者高速累计定时器。例如，指定 OUT ST0 时成为低速累计定时器，指定 OUTH ST0 时成为高速累计定时器。

(4) 低速定时器

(a) 关于低速定时器

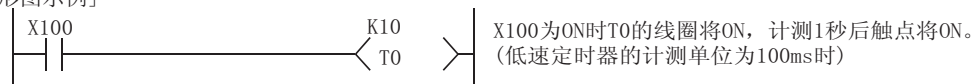
低速定时器是计测单位为 1 ~ 1000ms 的定时器。

定时器只有在线圈为 ON 的状态下有效。

定时器的线圈为 ON 时开始计测，“时间到”时触点将 ON。

定时器的线圈为 OFF 时，当前值变为 0，触点也将 OFF。

[梯形图示例]



[时序图]

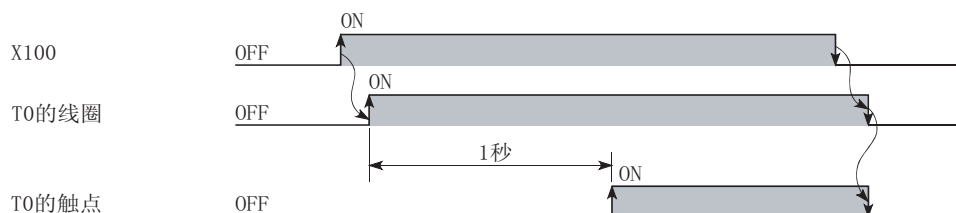


图 9.18 低速定时器的梯形图示例及时序图

(b) 计测单位

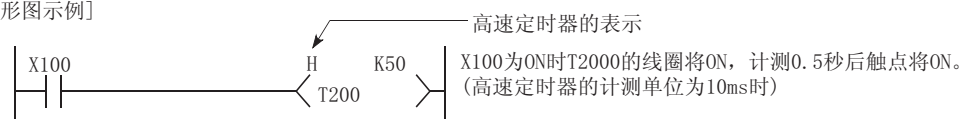
低速定时器的计测单位（时限）的缺省值为 100ms。
计测单位为 1 ~ 1000ms，可以以 1ms 为单位进行更改。
设置是在可编程控制器参数的可编程控制器系统设置中进行。

(5) 高速定时器

(a) 关于高速定时器

高速定时器是计测单位为 0.1 ~ 100ms 的定时器。
定时器只有在线圈为 ON 的状态下有效，附加符号“H”。
定时器的线圈为 ON 时开始计测，“时间到”时触点将 ON。
定时器的线圈为 OFF 时当前值变为 0，触点也将 OFF。

[梯形图示例]



[时序图]

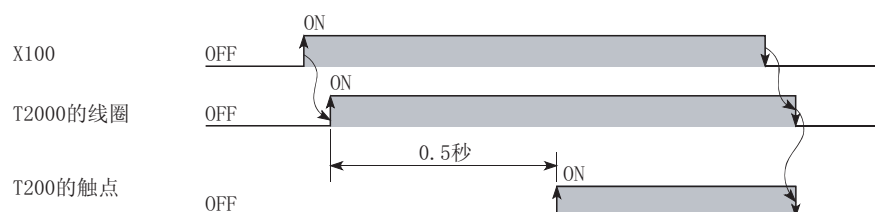


图 9.19 高速定时器的梯形图示例及时序图

(b) 计测单位

高速定时器的计测单位（时限）的缺省值为 10ms。
计测单位为 0.1 ~ 100ms，可以以 0.1ms 为单位进行更改。
设置是在可编程控制器参数的可编程控制器系统设置中进行。

(6) 累计定时器

(a) 关于累计定时器

累计定时器是计测线圈为 ON 状态时的时间的定时器。

定时器的线圈为 ON 时开始计测，“时间到”时触点将 ON。

即使定时器的线圈变为 OFF，也将保持当前值、触点的 ON/OFF 状态。

再次线圈为 ON 时，从保持的当前值开始重新计测。

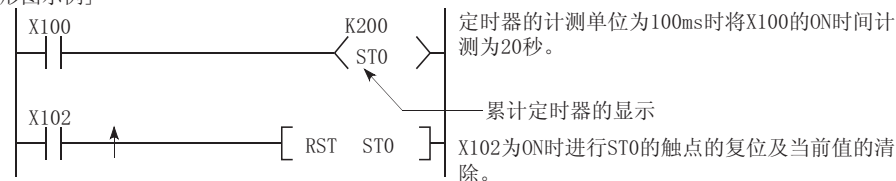
(b) 累计定时器的类型

累计定时器中有低速累计定时器及高速累计定时器这 2 种。

(c) 累计定时器的清除

当前值的清除及触点的 OFF 是通过 RST T₀ 指令进行的。

[梯形图示例]



[时序图]

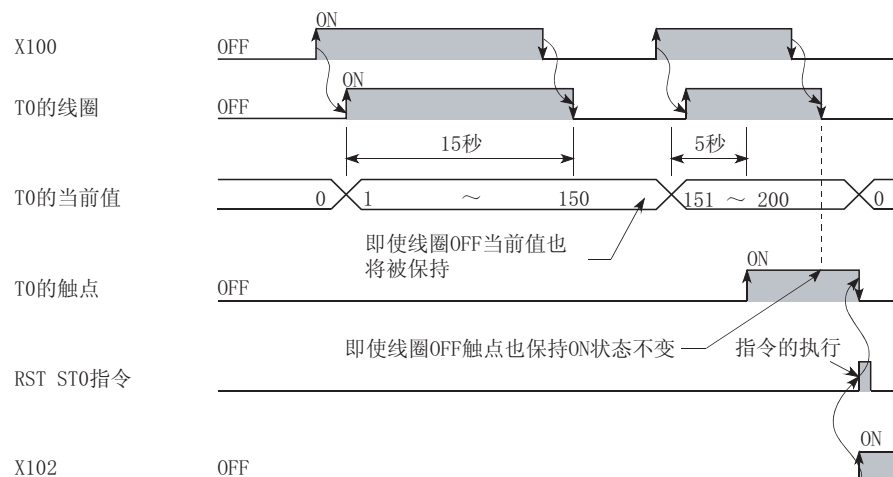


图 9.20 累计定时器的梯形图示例及时序图

(d) 计测单位

低速累计定时器与高速累计定时器的计测单位（时限）相同。

- 低速累计定时器：低速定时器
- 高速累计定时器：高速定时器

☒ 要点

使用累计定时器时，需要在可编程控制器的软元件设置中设置累计定时器的使用点数。

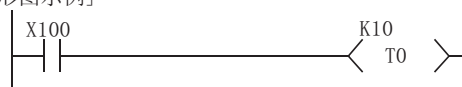
(7) 定时器的处理方法及精度

(a) 处理方法

执行 OUT T₀ 指令时执行定时器线圈的 ON/OFF、当前值的更新以及触点的 ON/OFF 处理。

不通过 END 处理进行定时器的当前值的更新及触点的 ON/OFF 处理。

[梯形图示例]



[执行OUT T₀指令时的处理内容]

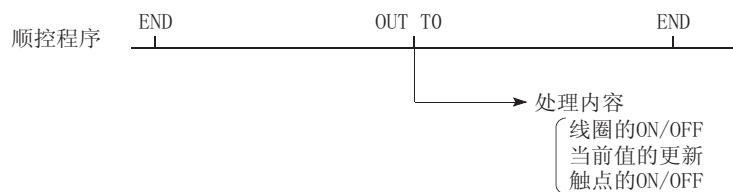


图 9.21 执行 OUT T₀ 指令时的处理内容

在执行 OUT T₀ 指令时将通过 END 指令计测的扫描时间的值加到当前值中。
执行 OUT T₀ 指令时定时器的线圈为 OFF 时，不进行当前值的更新。

```

graph LR
    L1 --- X100
    X100 --- H
    H --- K8
    K8 --- T0
    T0 --- L2

```

The timing chart illustrates the operation of a TON timer in a PLC. The signals shown are:

- 程序 (Program):** Shows the execution of the program with labels for OUT T0 and END处理 (END processing).
- X100的外部输入 (X100 External Input):** The input signal that starts ON and then turns OFF.
- CPU模块的X100 (CPU Module X100):** The internal signal path for the input.
- T0的线圈 (T0 Coil):** The output signal that turns ON when the timer starts.
- T0的触点 (T0 Contact):** The output signal that turns ON when the timer reaches its setpoint.
- 10ms的计测 (10ms Measurement):** The time base for the timer, shown as 10ms intervals.
- 通过END指令的计数 (Count of END Instructions):** The number of times the END instruction is executed, which corresponds to the number of scans.
- T0的当前值 (T0 Current Value):** The current value of the timer, which increases by 1 for each scan cycle where the input is ON.

The chart shows that the timer value increases by 1 for each scan cycle where the input is ON. The timer value is 0 at the start, and it increases to 10 when the input turns OFF. The timer then resets to 0 when the input turns ON again.

图 9.22 定时器的精度

获取输入 (X) 至输出为止的定时器的响应精度最大为 “2 扫描时间 + 定时器时限设置”。

(8) 使用定时器时的注意事项

使用定时器时的注意事项如下所示。

(a) 关于同一定时器的使用

在 1 个扫描中不能将同一个定时器通过多个 (OUT T□) 进行记述。

将同一个定时器通过多个 OUT T□ 进行记述时，由于执行各 OUT T□ 指令时进行定时器的当前值的更新，因此不能正常进行计测。

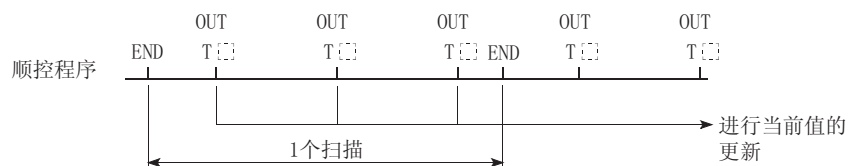


图 9.23 使用同一个定时器

(b) 设置值为 0 时

执行 OUT T□ 指令时触点将 ON。

(c) “时间到”后更改了设置值时

定时器发生“时间到”后，即使将设置值更改为大于当前值的值时，定时器将不动作而保持“时间到”状态不变。

9.2.9 计数器 (C)

(1) 关于计数器

计数器是对顺控程序中的输入条件的上升沿次数进行计数的软元件。

计数器值与设置值相同时将变为“计数到”，触点将 ON。

计数器为加法运算型。

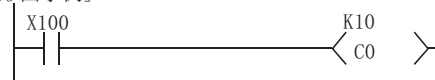
(2) 计数处理

(a) OUT C 指令时

执行 OUT C 指令时进行计数器的线圈的 ON/OFF、当前值的更新（计数值 +1）以及触点的 ON/OFF 处理。

不通过 END 处理进行计数器的当前值的更新及触点的 ON/OFF 处理。

[梯形图示例]



[执行 OUT C 指令时 (X100: OFF→ON 时) 的处理内容]

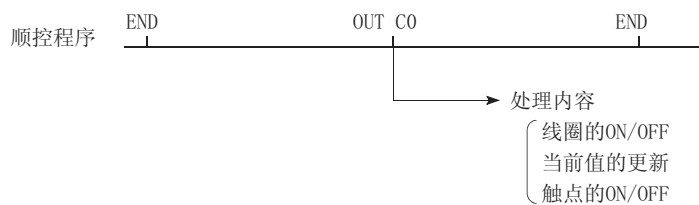


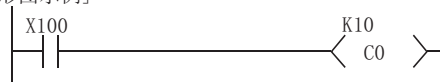
图 9.24 OUT C 指令的执行及处理内容

(b) 当前值的更新 (计数值 +1)

当前值的更新 (计数值 +1) 是在 OUT C 指令的上升沿时 (OFF → ON) 执行。

OUT C 指令为 OFF、OFF → ON 以及 ON → OFF 时, 不进行当前值的更新。

[梯形图示例]



[当前值的更新时机]

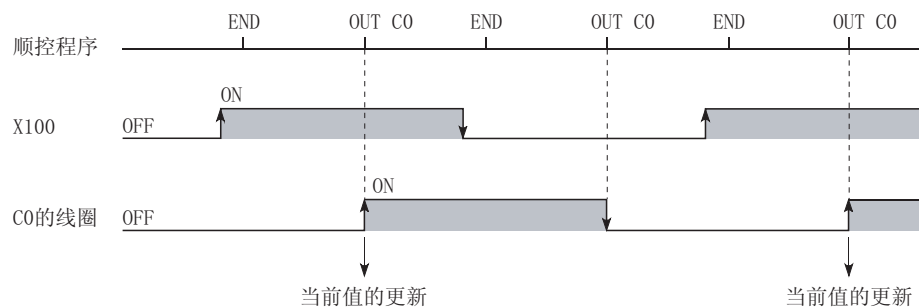


图 9.25 当前值的更新时机

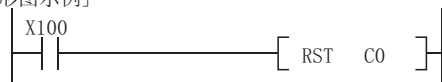
(c) 计数器的复位

即使 OUT C_□指令为 OFF 也不清除计数器的当前值。

计数器的当前值的清除 (复位) 与触点的 OFF 是通过 RST C_□指令执行的。

在执行 RST C_□指令的时点计数器值将被清除, 触点也将 OFF。

[梯形图示例]



[计数器的复位时机]

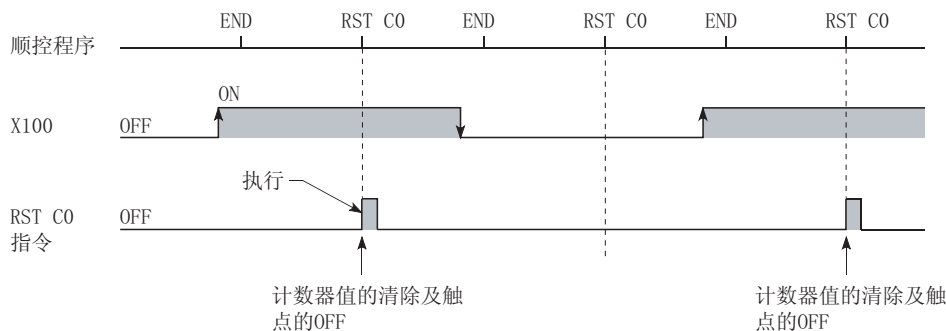


图 9.26 计数器的复位

1) 计数器复位时的注意事项

执行 RST C_□指令时 C_□的线圈也将 OFF。

执行 RST C_□指令后 OUT C_□指令的执行条件也为 ON 时, 执行 OUT C_□指令时 C_□的线圈将 ON, 执行当前值的更新 (计数值 +1)。

[梯形图示例]

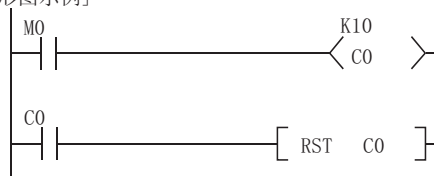
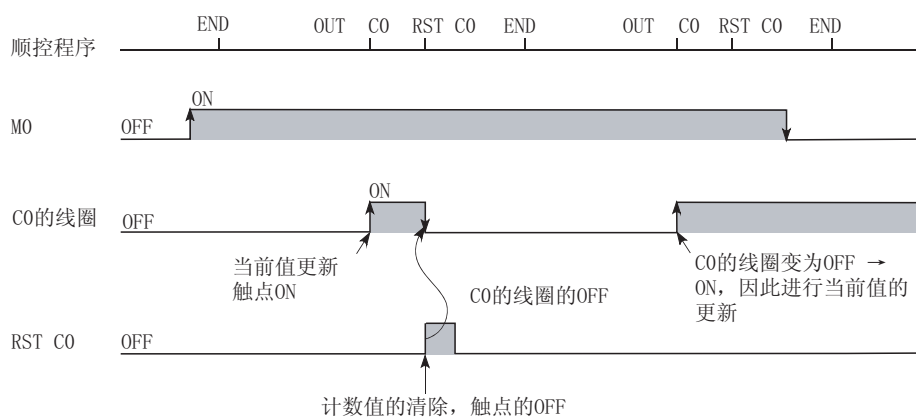


图 9.27 计数器复位的梯形图示例

在上述梯形图示例中, 通过 M0 的 OFF → ON 使 C0 的线圈为 ON, 当前值将被更新。

C0 为 “计数到” 时 C0 的触点将 ON, 通过执行 RST C0 指令 C0 的当前值将被清除。此时, C0 的线圈也将 OFF。

在下一个扫描中 M0 为 ON 的情况下, 执行 OUT C0 的指令时 C0 的线圈将变为 OFF → ON, 因此进行当前值的更新。(当前值将变为 1。)



与上述相对应，建议在 RST C0 指令的执行条件中插入 OUT C0 指令的执行条件的常闭触点，使 OUT C0 指令的执行条件 (M0) 为 ON 的期间 C0 的线圈不 OFF。

[更改梯形图示例]

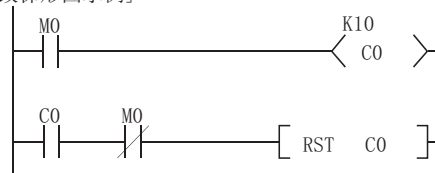


图 9.29 计数器复位的梯形图示例 (建议示例)

(d) 计数器的最大计数速度

计数器只有在输入条件的 ON/OFF 时间大于同一 OUT C_□ 指令的执行间隔时才可以计数。

计数器的最大计数速度可通过以下公式算出。

$$\text{最大计数速度 (Cmax)} = \frac{n}{100} \times \frac{1}{T} [\text{次/s}]$$

n: 任务 (%)^{*1}
T: OUT C_□ 指令的执行间隔

*1: 任务 (n) 是指，将计数输入信号的 ON/OFF 时间之比以百分比 (%) 表示。

$$\begin{aligned} \text{• } T1 \geq T2 \text{ 时 } n &= \frac{T1}{T1+T2} \times 100\% \\ \text{• } T1 < T2 \text{ 时 } n &= \frac{T2}{T1+T2} \times 100\% \end{aligned}$$

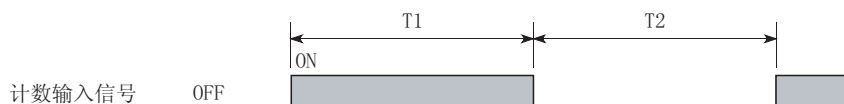


图 9.30 任务比

9.2.10 数据寄存器 (D)

(1) 关于数据寄存器

数据寄存器是可以存储数值数据 (-32768 ~ 32767 或者 0000H ~ FFFFH) 的存储器。

(2) 数据寄存器的位构成

(a) 位构成及读取、写入单位

数据寄存器 1 点由 16 位所构成，可以以 16 位为单位进行读取及写入。

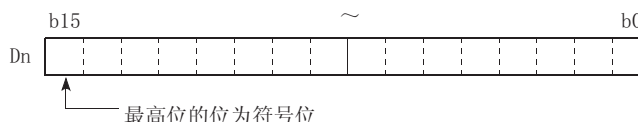


图 9.31 数据寄存器的位构成

☒ 要 点

数据寄存器是带符号使用的。

HEX(16 进制数) 时可以存储 0000H ~ FFFFH，但最高位的位为符号位，因此可指定的数值范围为 -32768 ~ 32767。

(b) 在 32 位指令中使用数据寄存器

在 32 位指令中使用数据寄存器时，Dn 及 Dn+1 将成为处理对象。

在顺控程序中指定的数据寄存器编号 (Dn) 为低位 16 位，在顺控程序中指定的数据寄存器编号 +1 的数据寄存器为高位 16 位。

例如，在 DMOV 指令中指定了 D12 时，D12 为低位 16 位，D13 为高位 16 位。

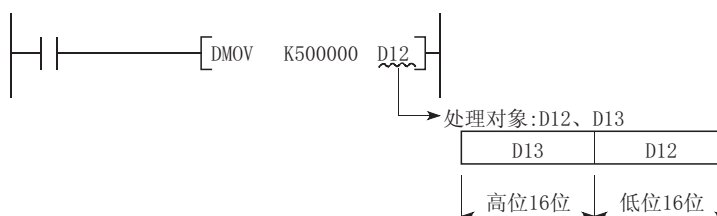


图 9.32 通过 32 位指令进行的数据传输及存储目标

在数据寄存器 2 点中可以存储，-2147483648 ~ 2147483647 或者 0H ~ FFFFFFFFH 的数据。(32 位构成时的最高位的位为符号位。)

(3) 关于存储数据的保持

数据寄存器中存储的数据将被一直保持，直到其它的数据被存储为止。

数据寄存器中存储的数据在进行了可编程器的电源 OFF 或者 CPU 模块的复位时将被初始化。

9.2.11 链接寄存器 (W)

(1) 关于链接寄存器

链接寄存器是指，将 MELSECNET/H 模块等的链接寄存器 (LW) 的数据刷新到 CPU 模块中的 CPU 模块侧的存储器。

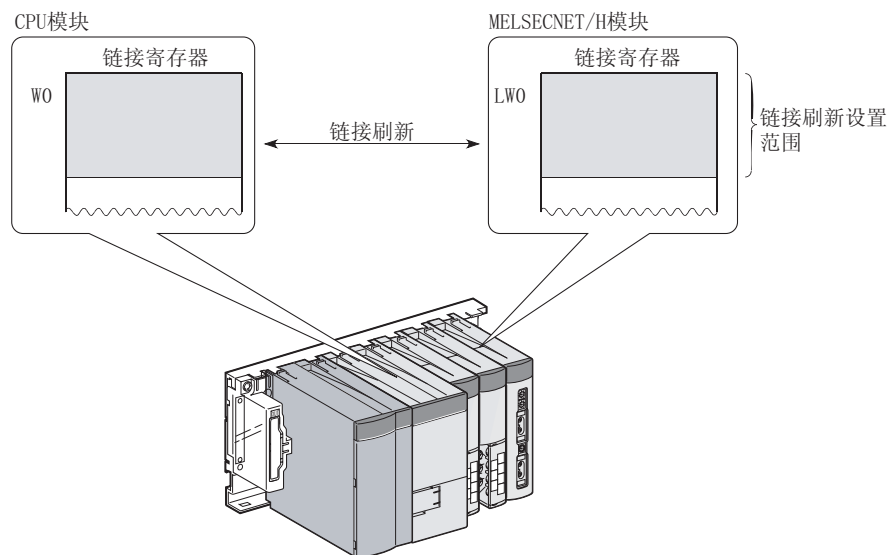


图 9.33 链接刷新

链接寄存器是可以存储数值数据 ($-32768 \sim 32767$ 或者 $0000H \sim FFFFH$) 的存储器。

(2) 链接寄存器的位构成

(a) 位构成及读取、写入单位

链接寄存器 1 点由 16 位所构成，可以以 16 位为单位进行读取及写入。

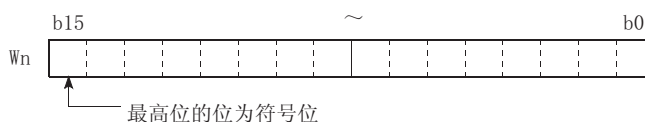


图 9.34 链接寄存器的位构成

☒ 要 点

1. 链接寄存器是带符号使用的。
HEX(16 进制数) 时可以存储 $0000H \sim FFFFH$ ，但最高位的位为符号位，因此可指定的数值范围为 $-32768 \sim 32767$ 。
2. 在 MELSECNET/H 模块等中未使用的范围的链接寄存器可以作为数据寄存器的代用品使用。

(b) 在 32 位指令中使用链接寄存器

在 32 位指令中使用链接寄存器时, 连续的 2 点链接寄存器 (W_n 与 W_{n+1}) 将成为处理对象。

在顺控程序中指定的链接寄存器编号 (W_n) 为低位 16 位, 在顺控程序中指定的链接寄存器编号 +1 的链接寄存器为高位 16 位。

例如, 在 DMOV 指令中指定了 $W12$ 时, $W12$ 为低位 16 位, $W13$ 为高位 16 位。

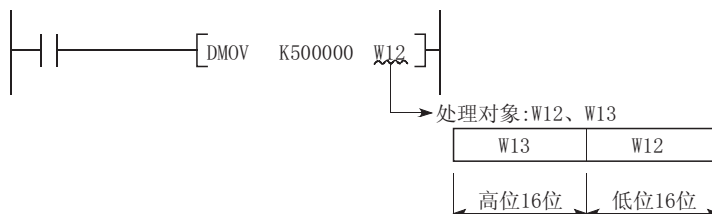


图 9.35 通过 32 位指令进行的传输及存储目标

在链接寄存器 2 点中可以存储 $-2147483648 \sim 2147483647$ 或者 $0H \sim FFFFFFFFH$ 的数据。(32 位构成时的最高位的位为符号位。)

(3) 关于存储数据的保持

链接寄存器中存储的数据将被一直保持, 直至其它的数据被存储为止。

链接寄存器中存储的数据在进行了可编程器的电源 OFF 或者 CPU 模块的复位时将被初始化。

☒ 要 点

MELSECNET/H 模块内的链接寄存器为 16384 点, 但 CPU 模块内的链接寄存器的缺省值为 2048 点。

使用上述所示的点数以后的链接寄存器时, 应在可编程控制器参数的软元件设置中更改链接寄存器的点数。

(4) 在网络系统中使用时

在网络系统中使用时，需要进行网络参数的设置。

对于设置网络参数中未设置的链接寄存器，可以作为数据寄存器的代用品使用。

备注

关于网络参数，请参阅以下手册。

☞ Q 系列 MELSECNET/H 网络系统参考手册 (PLC 网络篇)

9.2.12 链接特殊寄存器 (SW)

(1) 关于链接特殊寄存器

链接特殊寄存器是存储 CC-Link Safety 主站模块以及 MELSECNET/H 模块的通信状态、异常内容的寄存器。

数据链接时的信息是以数值被存储的，因此通过监视链接特殊寄存器，可以调查异常位置及原因。

(2) 链接特殊寄存器点数

链接特殊寄存器的点数如表 9.4 所示。

表 9.4 各 CPU 模块的链接特殊寄存器的点数

CPU 模块	链接特殊寄存器点数
安全 CPU	1536 点 (SW0 ~ 5FF)。 CC-Link Safety 主站模块以及 MELSECNET/H 模块各 512 点。 对链接特殊寄存器可以进行如下图所示的分配。 SW0 { SW1FF SW200 { SW3FF SW400 { SW5FF 链接特殊寄存器 第1个的网络模块用 第2个的网络模块用 第3个的网络模块用 } 512 点 } 512 点 } 512 点 } 1536 点

备注

关于链接特殊寄存器的详细内容，请参阅以下手册。

☞ CC-Link Safety 主站模块用户手册 (详细篇)

☞ Q 系列 MELSECNET/H 网络系统参考手册 (PLC 网络篇)

9.3 内部系统软元件

内部系统软元件是系统用的软元件。
内部系统软元件的分配 / 容量是事先固定的，用户无法更改。

9.3.1 特殊继电器 (SM)

(1) 关于特殊继电器

特殊继电器是存储 CPU 模块的状态（故障诊断、系统信息等）的继电器。

(2) 特殊继电器的分类

特殊继电器可以根据用途按表 9.5 所示进行分类。

表 9.5 特殊继电器的分类一览

分类	特殊继电器
诊断信息	SM0 ~ 99
系统信息	SM200 ~ 399
系统时钟 / 系统计数器	SM400 ~ 499
安全 CPU	SM560 ~ 599
引导运行	SM600 ~ 699
指令相关	SM700 ~ 799
CC-Link Safety	SM1000 ~ 1299

(3) 实现安全功能的程序中可使用的特殊继电器

在实现安全功能的程序中只能使用 SM1000 ~ SM1299。

备注

关于特殊继电器的详细内容，请参阅附录 1。

9.3.2 特殊寄存器 (SD)

(1) 关于特殊寄存器

特殊寄存器是存储 CPU 模块的状态（故障诊断、系统信息等）的寄存器。

(2) 特殊寄存器的分类

特殊寄存器可以根据用途按表 9.6 所示进行分类。

表 9.6 特殊寄存器的分类一览

分类	特殊寄存器
诊断信息	SD0 ~ 99
系统信息	SD200 ~ 399
系统时钟 / 系统计数器	SD400 ~ 499
扫描信息	SD500 ~ 559
安全 CPU	SD560 ~ 599
存储器	SD600 ~ 699
CC-Link Safety	SD1000 ~ 1299

(3) 实现安全功能的程序中可使用的特殊寄存器

在实现安全功能的程序中只能使用 SD1000 ~ SD1299。

备注

关于特殊寄存器的详细内容，请参阅附录 2。

9.4 嵌套 (N)

(1) 关于嵌套

嵌套是在主控制指令 (MC 指令、MCR 指令) 中使用的, 将动作条件通过嵌套进行编程的软元件。

(2) 在主控制指令中的指定方法

主控制指令是指, 通过梯形图的公共母线的开闭, 创建高效的梯形图切换的顺控程序的指令。

嵌套是从嵌套的外侧按从小到大 (从 N0 至 N14 的顺序) 的编号顺序指定。

关于嵌套的使用方法, 请参阅以下手册。

☞ QSCPU 编程手册 (公共指令篇)

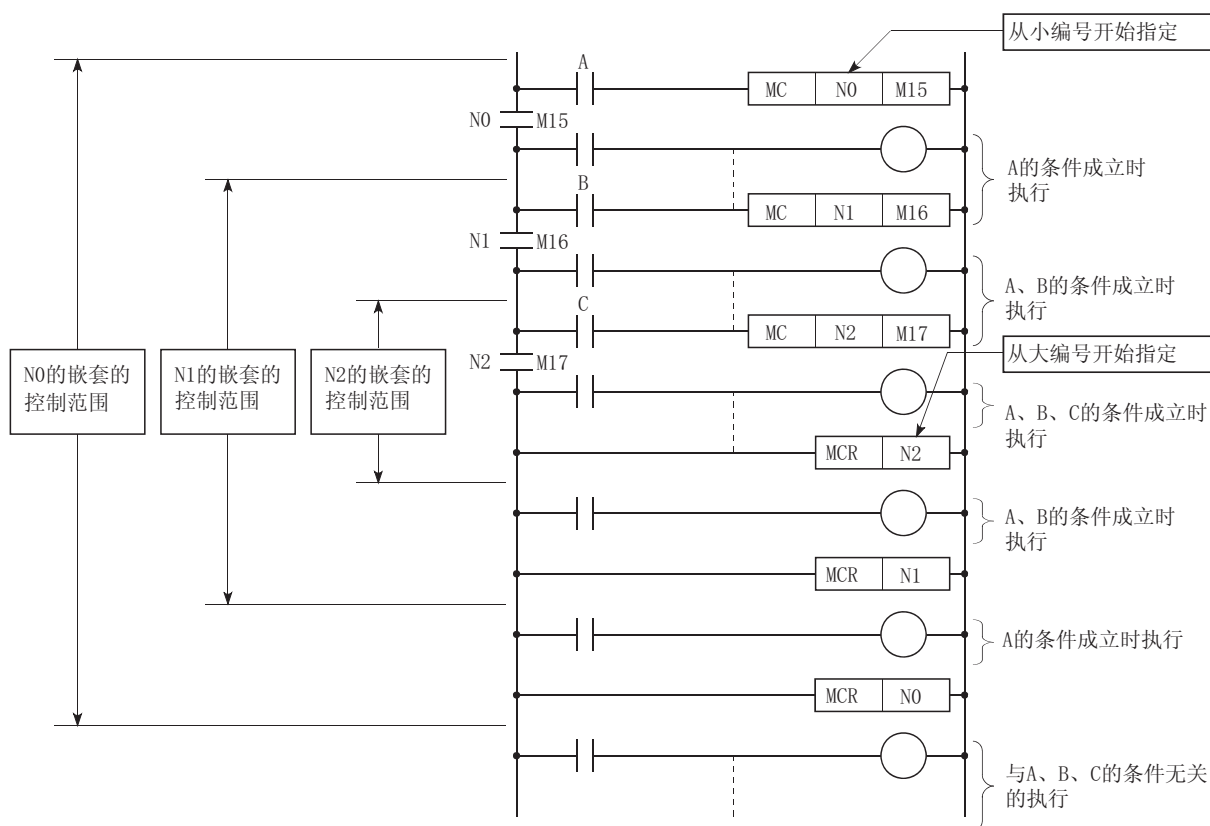


图 9.36 使用了嵌套的程序示例

9.5 常数

9.5.1 10 进制数常数 (K)

(1) 关于 10 进制数常数

10 进制数常数是在顺控程序中指定 10 进制数数据的软元件。

在顺控程序中，以 K□□ (例：K1234) 进行指定。

CPU 模块内部是以 2 进制数 (BIN) 存储的。(☞ 3.7.1 节)

(2) 指定范围

10 进制数的指定如下所示。

- 使用字数据 (16 位) 时 K-32768 ~ 32767
- 使用 2 字数据 (32 位) 时 K-2147483648 ~ 2147483647

☒ 要 点

最高位的位为符号位。

9.5.2 16 进制数常数 (H)

(1) 关于 16 进制数常数

16 进制数常数是在顺控程序中指定 16 进制数或者 BCD 数据的软元件。

(以 BCD 进行数据指定时，将 16 进制数的各位按 0 ~ 9 进行指定。)

在顺控程序中，以 H□□ (例：H1234) 进行指定。(☞ 3.7.2 节)

(2) 指定范围

16 进制数的设置范围如下所示。

- 使用字数据 (16 位) 时 H0 ~ FFFF
(BCD 数据时为 H0 ~ 9999)
- 使用 2 字数据 (32 位) 时 H0 ~ FFFFFFFF
(BCD 数据时为 H0 ~ 99999999)

☒ 要 点

最高位的位为符号位。

第 10 章 CPU 模块的处理时间

本章介绍 CPU 模块的处理时间有关内容。

10.1 扫描时间

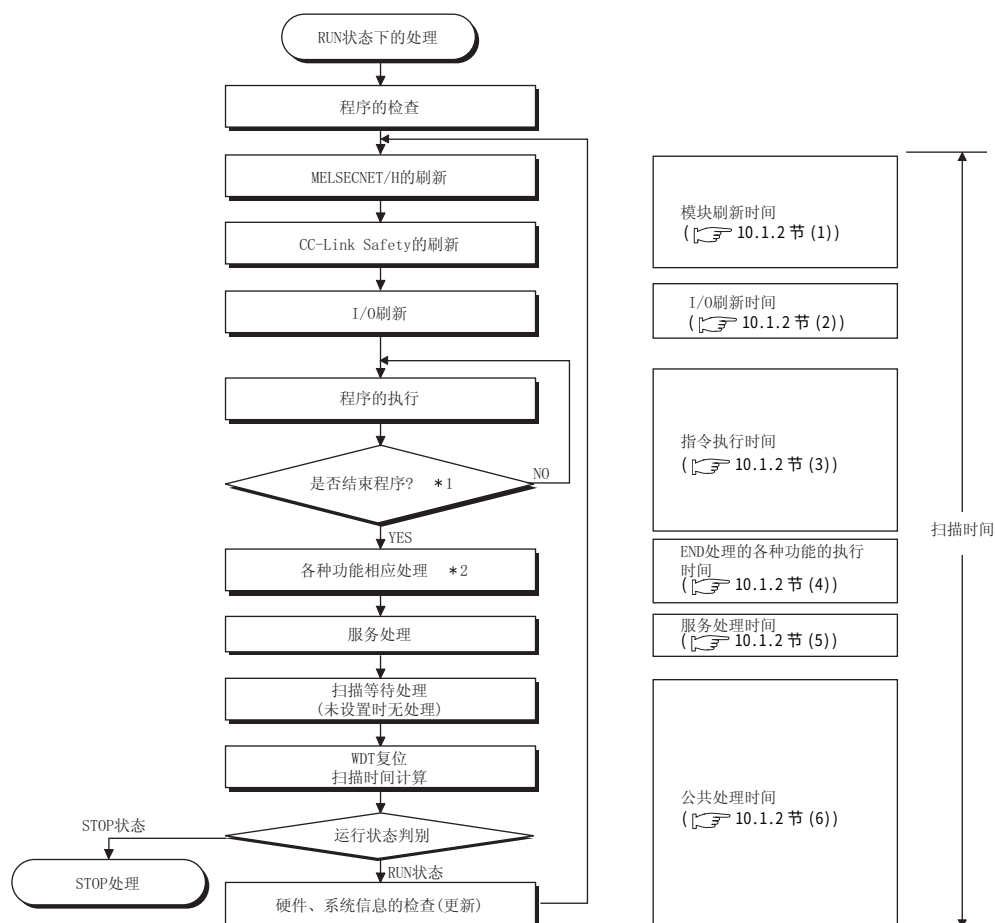
本节介绍 CPU 模块的扫描时间的构成及处理时间有关内容。

10.1.1 扫描时间的构成及计算公式

(1) 扫描时间的构成

CPU 模块的扫描时间的构成如下所示。

CPU 模块在运行状态时循环执行以下的处理。



*1：程序结束表示执行了 END、S.QSABORT 指令的时机。

*2：表示日历更新、出错解除。

图 10.1 安全 CPU 的扫描时间的构成

(2) 扫描时间的计算

扫描时间的计算公式如下。

$$SM = Tru + Tio + Tie + Tend + Ts + Tc \text{ (ms)}$$

- SM : 扫描时间
- Tru : 模块刷新时间
- Tio : I/O 刷新时间
- Tie : 指令执行时间
- Tend : END 处理的各种功能的执行时间
- Ts : 服务处理时间
- Tc : 公共处理时间

10.1.2 扫描时间相关要素的处理时间

以下介绍 10.1.1 节中所示的处理以及执行时间的计算方法有关内容。

(1) 模块刷新时间

模块刷新时间是指，执行网络参数中设置的 MELSECNET/H 以及 CC-Link Safety 的自动刷新的合计时间。

(a) MELSECNET/H 的刷新

是在 MELSECNET/H 模块的链接软件与 CPU 模块的软件之间进行数据刷新的时间。
MELSECNET/H 的刷新时间 (Tmnet) 的计算公式如下。

$$T_{mnet} = 1.85 \times \text{刷新字数} + 1000 (\mu s)$$

关于刷新字数的有关内容，请参阅以下手册。

☞ Q 系列 MELSECNET/H 网络系统参考手册 (PLC 网络篇)

(b) CC-Link Safety 的自动刷新

是在 CC-Link Safety 主站模块与 CPU 模块之间进行数据刷新的时间。

关于 CC-Link Safety 的自动刷新时间，请参阅以下手册。

☞ CC-Link Safety 主站模块用户手册 (详细篇)

(2) I/O 刷新时间

I/O 刷新时间是指，与 CC-Link Safety 主站模块、MELSECNET/H 模块之间的 I/O 数据的刷新时间。

I/O 刷新时间 (Tio) 的计算公式如下。

$$T_{io} = I/O \text{ 点数} \times 0.224 + 310 (\mu s)$$

(3) 指令执行时间

指令执行时间是 CPU 模块中执行的程序中使用各指令的处理时间的合计时间。

关于各指令的处理时间，请参阅以下手册。

☞ QSCPU 编程手册 (公共指令篇)

(4) END 处理的各种功能的执行时间

END 处理的各种功能的执行时间是时钟数据的读取、出错解除所需时间的合计时间。

(a) 时钟数据的读取处理时间

是进行了时钟数据读取请求 (SM213 为 ON) 时, 用于通过 END 处理读取时钟数据的时间。

表 10.1 时钟数据的读取处理时间

CPU 模块	有时钟数据读取请求
QS001CPU	0.02ms

(b) 出错解除处理

是 SM50(出错解除) 的上升沿 (从 OFF 变为 ON) 时, 对 SD50 中存储的继续运行出错进行出错解除的时间

表 10.2 出错解除处理时间

CPU 模块	出错解除时间
QS001CPU	0.13ms

(5) 服务处理时间

服务处理是与 GX Developer 的通信处理。

- 通过 GX Developer 进行监视

服务处理时间是通过 GX Developer 进行监视时的处理时间。

表 10.3 通过 GX Developer 进行监视的处理时间

功能	QS001CPU
程序的可编程控制器读取 *1	1.4ms
软元件监视 *2	1.0ms
运行中写入 *3	4.0ms
操作 / 故障历史记录显示 *4	6.1ms

*1: 是从程序存储器中读取 8k 步的程序时所需的时间。

*2: 是在登录监视中设置了 32 点时所需的时间。

*3: 是添加了 100 步的梯形图时所需的时间。

*4: 是指定了 [All log(全部记录)] 时显示更新所需的时间。

(6) 公共处理时间

是系统中处理的 CPU 模块的公共处理时间。

公共处理时间的值如表 10.4 所示。

表 10.4 公共处理时间

CPU 模块	公共处理时间
QS001CPU	6.2 ~ 8.2ms

10.1.3 扫描时间的延长原因

本项中所示的功能或者操作是使 CPU 模块的扫描时间延长的原因。
执行本项中所示的功能或者操作时，应将本项中所示的值加到 10.1.2 节中算出的值中。

(1) 梯形图模式下的运行中写入的执行

执行运行中写入时，扫描时间将延长表 10.5 中所示的值。

表 10.5 执行了运行中写入时的延长时间

CPU 模块型号	运行中写入用预留容量 (步)	
	无更改	重新设置
QS001CPU	最大 1.7ms	最大 36ms

(2) 使扫描时间延长的功能

使用以下功能时也会延长扫描时间。

- 系统监视
- 常规数据处理

(3) 以固定周期进行自诊断

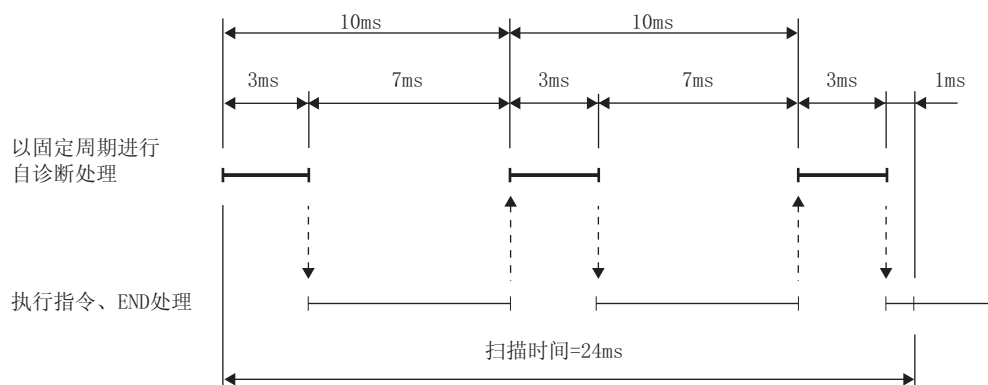
在 CPU 模块中，存在有与扫描无关以固定周期 (10ms) 进行自诊断的处理。

表 10.6 以固定周期进行自诊断处理的处理时间

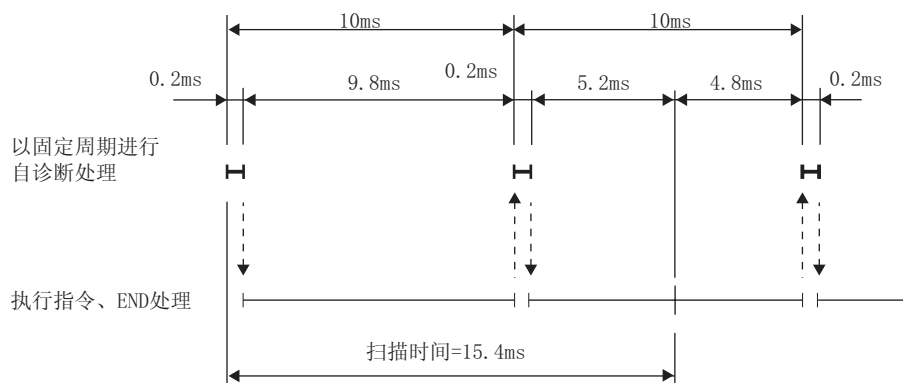
CPU 模块型号	处理时间
QS001CPU	0.2 ~ 3.0ms

因此，扫描时间根据以固定周期进行自诊断处理的处理时间而有所变化。

通过 10.1.1 节的计算公式算出的扫描时间为 15ms 时，根据以固定周期进行自诊断处理的扫描时间的变化如图 10.2 所示。



(a) 以固定周期进行自诊断处理的处理时间为最大时



(b) 以固定周期进行自诊断处理的处理时间为最小时

图 10.2 根据以固定周期进行自诊断处理的处理时间的扫描时间的变化

10.2 其它处理时间

本节介绍除 10.1 节中介绍的以外的处理时间的有关内容。

(1) 恒定扫描的精度

恒定扫描的精度如表 10.7 所示。

表 10.7 恒定扫描的精度

CPU 模块	恒定扫描的精度
QS001CPU	2ms

有监视：表示通过连接 GX Developer 进行监视的状态。

无监视：表示未通过 GX Developer 进行监视的状态。

备忘录

[illegible]

第 11 章 将程序写入 CPU 模块中的步骤

本章介绍将通过 GX Developer 创建的程序写入到 CPU 模块中的步骤的有关内容。

关于 CPU 模块的启动步骤，本手册中未加以记载。

关于 CPU 模块的启动步骤，请参阅以下手册。

☞ QSCPU 用户手册（硬件设计 / 维护点检篇）

11.1 创建程序时的确认事项

在 CPU 模块中创建程序时，需要预先确定程序容量、使用软元件点数等。

(1) 程序容量的确认

确认在 CPU 模块中可执行的程序容量（14k 步）以内，能否存储程序。

☞ 5.3.3 节

(2) 软元件的用途及点数的设置

确认程序中使用的软元件的用途及点数。（☞ 第 9 章）

(3) 引导运行的确认

在测试模式下进行引导运行时，进行可编程控制器参数的引导文件设置。

（在安全模式下与可编程控制器参数的引导文件设置无关，进行引导运行。）

☞ 5.1.4 节，☞ 11.3 节

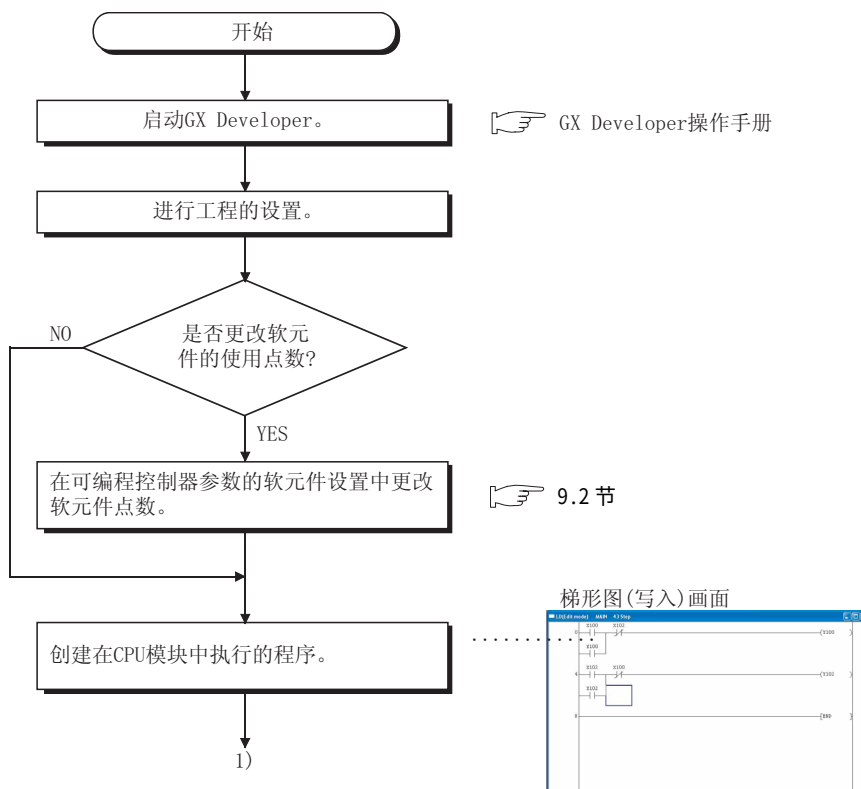
11.2 程序的写入步骤

本节介绍将通过 GX Developer 创建的参数、程序写入到 CPU 模块中的步骤。

以下介绍将程序写入到程序存储器 (5.1.2 节) 中的步骤。

在测试模式下将程序存储到内置 ROM 中进行引导运行时, 应在执行了以下的步骤后执行 11.3 节中所示的步骤。

在以下步骤中, □ 表示 GX Developer 方面的操作项目, ▢ 表示 CPU 模块方面的操作项目。



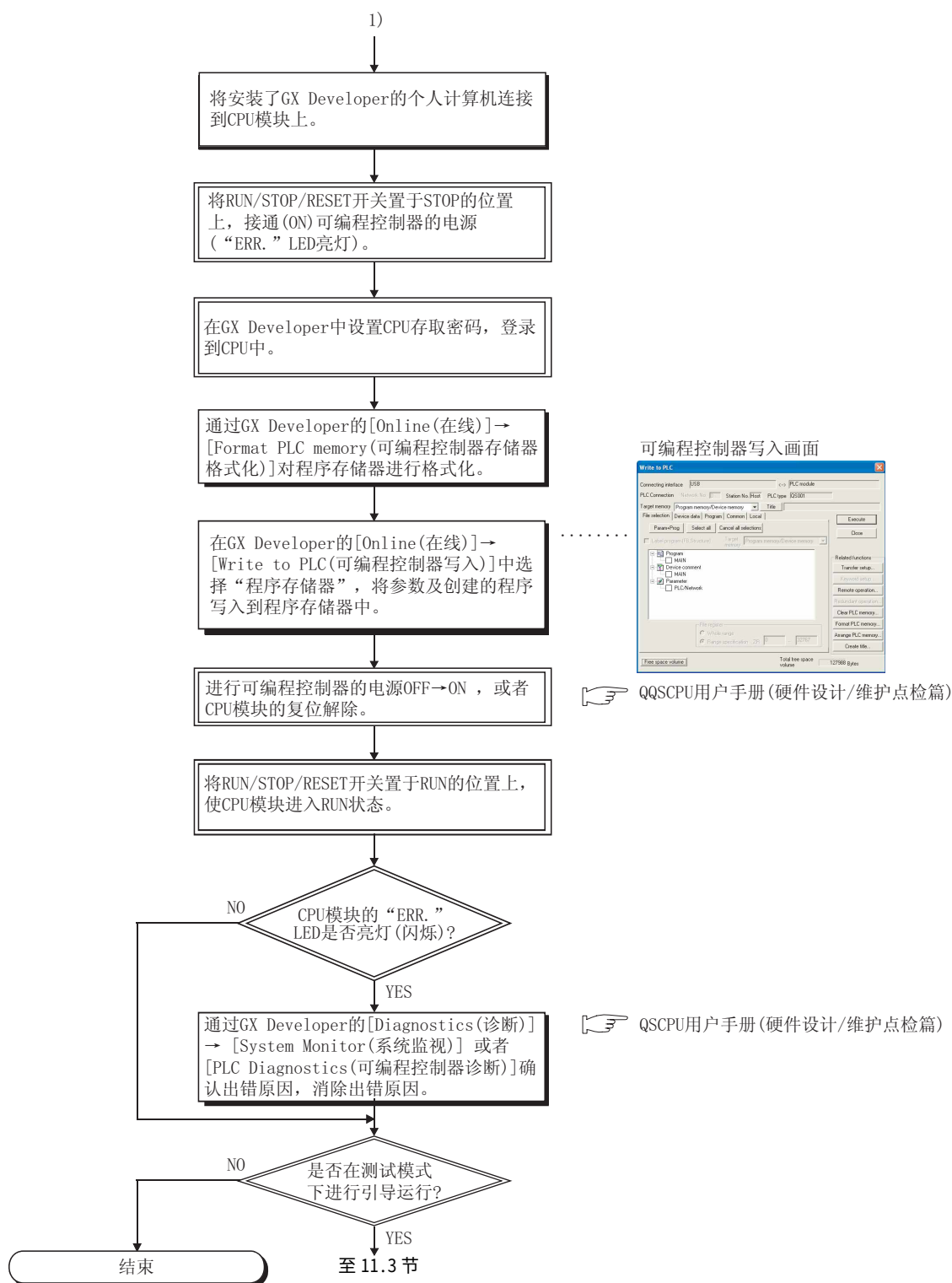


图 11.1 程序写入的流程

11.3 引导运行的步骤

本节介绍在测试模式下进行引导运行时的步骤。

在以下步骤中，□表示 GX Developer 方面的操作项目，▣表示 CPU 模块方面的操作项目。

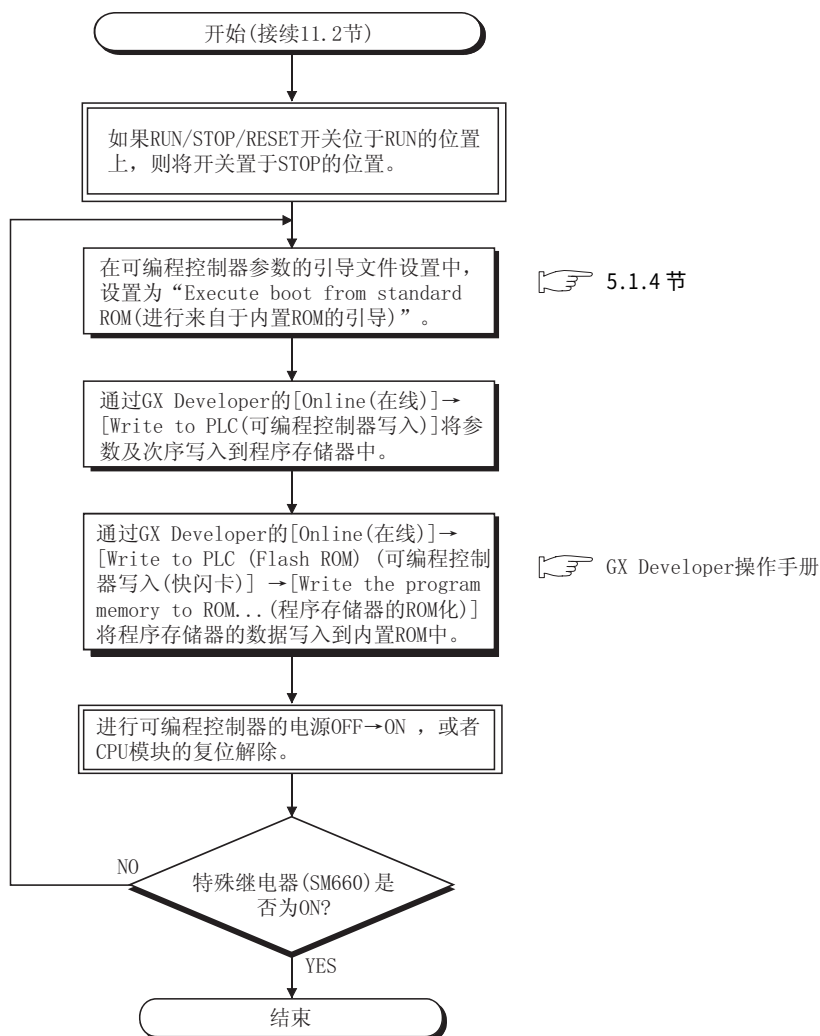


图 11.2 引导运行的流程

附录

附录 1 特殊继电器一览

特殊继电器 SM 是可编程控制器内部确定了规格的内部继电器。
因此不能象通常的内部继电器那样在顺控程序中使用。
但是，根据需要可以进行用于 CPU 模块控制的 ON/OFF。
一览表中的各项目的含义如下所示。

附表 1 特殊继电器一览表的阅读方法

项目	项目说明
编号	• 显示特殊继电器的编号。
名称	• 显示特殊继电器的名称。
内容	• 显示特殊继电器的有关内容。
详细内容	• 显示特殊继电器的有关详细内容。
设置侧（设置时间）	• 说明设置侧及系统侧中设置时的时间有关内容。 < 设置侧 > S : 在系统侧设置。 U : 在用户侧（顺控程序或者通过 GX Developer 的测试操作）设置。 S/U : 在系统侧 / 用户侧均设置。 < 设置时间 > 仅在系统侧设置时显示设置时间。 每次 END: 每次 END 处理时设置。 初始: 仅在初始化（电源 ON、STOP → RUN 等）时设置。 状态变化: 仅在状态发生了变化时设置。 发生出错: 在发生了出错时设置。 执行指令: 在执行指令时设置。 请求时: 仅在有用户的请求时（通过 SM 等）设置。

关于以下项目的详细内容，请参阅以下手册。
· 网络相关 → CC-Link Safety 主站模块用户手册（详细篇）
 → Q 系列 MELSECNET/H 网络系统参考手册（PLC 网络篇）

☒ 要 点

在实现安全功能的程序中，只能使用 SM1000 ～ SM1299。
除 SM1000 ～ SM1299 以外的特殊继电器不能在实现安全功能的程序中使用。

(1) 诊断信息

附表 2 特殊继电器

编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU
SM0	诊断出错	OFF : 无出错 ON : 有出错	• 通过诊断检测出出错时将 ON。 (也包括报警器 ON 时。) • 以后变为正常时也保持 ON 状态不变。	S(发生出错)	QS
SM1	自诊断出错	OFF : 无自诊断出错 ON : 有自诊断出错	• 通过自诊断检测出出错时将 ON。 (不包括报警器 ON 时。) • 以后变为正常时也保持 ON 状态不变。	S(发生出错)	
SM5	出错公共信息	OFF : 无出错公共信息 ON : 有出错公共信息	• SM0 为 ON 时, 如果有出错公共信息将 ON。	S(发生出错)	
SM16	出错个别信息	OFF : 无出错个别信息 ON : 有出错个别信息	• SM0 为 ON 时, 如果有出错个别信息将 ON。	S(发生出错)	
SM50	出错解除	OFF → ON: 出错解除	• 执行出错解除动作。	U	
SM51	电池电量不足锁存	OFF : 正常 ON : 电池不足	• CPU 模块的电池电压低于规定值时将 ON。 • 以后即使电池电压正常也将保持 ON 状态不变。 • 与 “BAT” LED 同步。	S(发生出错)	
SM52	电池电量不足	OFF : 正常 ON : 电池不足	• 与 SM51 相同, 但以后电池电压恢复正常时将 OFF。	S(发生出错)	
SM53	检测出 AC DOWN	OFF : 无 AC DOWN ON : 有 AC DOWN	• 使用 AC 电源模块时发生了 20ms 以内的瞬时掉电时将 ON。 电源 OFF → ON 时将被复位。	S(发生出错)	
SM56	运算出错	OFF : 正常 ON : 有运算出错	• 发生了运算出错时将 ON。 • 以后即使变为正常时也将保持 ON 状态不变。	S(发生出错)	
SM61	模块校验出错	OFF : 正常 ON : 有出错	• I/O 模块在接通电源时与登录的状态不相同时将 ON。 • 即使以后恢复正常时也将保持 ON 状态不变。	S(发生出错)	
SM62	检测出报警器	OFF : 未检测出 ON : 检测出	• 如果有 1 个报警器 F 为 ON 则 SM62 将 ON。	S(执行指令)	

(2) 系统信息

附表 3 特殊继电器

编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU
SM203	STOP 触点	STOP 状态	• STOP 状态时将 ON。	S(状态变化)	QS
SM213	时钟数据读取请求	OFF : 无处理 ON : 读取请求	• 本继电器为 ON 时将时钟数据以 BCD 值读取到 SD210 ~ SD213 中。	U	
SM232	ROM 写入次数	OFF : 写入次数范围内 ON : 写入次数溢出	• ROM 写入次数超过 10 万次时将 ON。	S(发生出错)	

(3) 系统时钟 / 计数器

附表 4 特殊继电器

编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU	
SM400	常时 ON	ON  OFF	• 常时 ON。	S(每次 END)	QS	
SM401	常时 OFF	ON  OFF	• 常时 OFF。	S(每次 END)		
SM402	运行后仅 1 个扫描 ON	ON  OFF	• 运行后仅 1 个扫描 ON。	S(每次 END)		
SM403	运行后仅 1 个扫描 OFF	ON  OFF	• 运行后仅 1 个扫描 OFF。	S(每次 END)		
SM410	0.1 秒时钟		• 每隔一定时间反复 ON/OFF。 • 可编程控制器的电源 ON 或者 CPU 模块的复位时从 OFF 状态启动。	S(状态变化)		
SM411	0.2 秒时钟					
SM412	1 秒时钟					
SM413	2 秒时钟					
SM414	2n 秒时钟		• 按SD414中指定的时间(单位:秒)反复进行ON/OFF 的继电器。 • 可编程控制器的电源 ON 或者 CPU 模块的复位时从 OFF 状态启动。	S(状态变化)		

(4) 安全 CPU

附表 5 特殊继电器

编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU
SM560	测试模式标志	OFF : 测试模式以外 ON : 测试模式	- 以测试模式运行时将 ON。 - 以测试模式以外 (安全模式、安全模式 (再启动等待)) 的模式运行时将 OFF。	S(状态变化)	QS
SM561	测试模式连续 RUN 允许时间设置	OFF : 设置时间内 ON : 设置时间溢出	超出了参数中设置的测试模式连续 RUN 允许时间时将 ON。	S(发生出错)	

(5) 引导运行

附表 6 特殊继电器

编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU
SM660	引导运行	OFF : 程序存储器执行 ON : 引导运行中	(测试模式时) - 在进行来自于内置 ROM 的引导运行过程中将 ON。 - 未进行来自于内置 ROM 的引导运行时将 OFF。 (安全模式时) - 常时 ON。	S(初始)	QS

(6) 指令相关

附表 7 特殊继电器

编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU
SM722	BIN、DBIN 指令出错禁止标志	OFF : 检测出错 ON : 不检测出错	在 BIN、DBIN 指令中不希望出现 “ OPERATION ERROR ” 时使 SM722 为 ON。	U	QS

(7) CC-Link Safety

附表 8 特殊继电器

编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU
SM1004	安全站刷新通信状态 (第 1 个 CC-Link Safety 主站模块)	OFF : 正常 ON : 有通信异常站	存储安全站刷新通信状态。(SD1004 ~ SD1007 中存储各站的状态)	S(状态变化)	QS
SM1204	安全站刷新通信状态 (第 2 个 CC-Link Safety 主站模块)	OFF : 正常 ON : 有通信异常站	存储安全站刷新通信状态。(SD1204 ~ SD1207 中存储各站的状态)	S(状态变化)	

附录 2 特殊寄存器一览

特殊寄存器 SD 是可编程控制器内部确定了规格的内部寄存器。
因此不能象通常的内部寄存器那样在顺控程序中使用。
但是，根据需要可以进行用于 CPU 模块及远程 I/O 模块控制的数据写入。
特殊寄存器中存储的数据在无特别指定的情况下是以 BIN 值存储。
一览表中的各项目的含义如下所示。

附表 9 特殊寄存器一览表的阅读方法

项目	项目说明
编号	• 显示特殊寄存器的编号。
名称	• 显示特殊寄存器的名称。
内容	• 显示特殊寄存器的有关内容。
详细内容	• 显示特殊寄存器的有关详细内容。
设置侧 (设置时间)	• 说明设置侧及系统侧中设置时的时间有关内容。 < 设置侧 > S : 在系统侧设置。 U : 在用户侧 (通过顺控程序或者 GX Developer 的测试操作) 设置。 S/U : 在系统侧 / 用户侧均设置。 < 设置时间 > 仅在系统侧设置时显示设置时间。 每次 END: 每次 END 处理时设置。 初始: 仅在初始化 (电源 ON、STOP → RUN 等) 时设置。 状态变化: 仅在状态发生了变化时设置。 发生出错: 在发生了出错时设置。 执行指令: 在执行指令时设置。 请求时: 仅在有用户的请求时 (通过 SM 等) 设置。 ROM 化时: 在 ROM 化时设置。

关于以下项目的详细内容，请参阅以下手册。
· 网络相关 → CC-Link Safety 主站模块用户手册 (详细篇)
→ Q 系列 MELSECNET/H 网络系统参考手册 (PLC 网络篇)

☒ 要 点

在实现安全功能的程序中，只能使用 SD1000 ~ SD1299。
除 SD1000 ~ SD1299 以外的特殊寄存器不能在实现安全功能的程序中使用。

(1) 诊断信息

附表 10 特殊寄存器

编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU
SD0	诊断出错	诊断出错代码	- 以 BIN 代码存储通过诊断检测出出错时的出错代码。 - 与故障历史记录的最新信息的内容相同。	S(发生出错)	
SD1	诊断出错发生 时间	诊断出错发生 时间	SD0 的数据以 2 位的 BCD 代码存储更新的年 (公历、低 2 位)、月。 b15 ~ b8 b7 ~ b0 年 (0~99) 月 (1~12) 例) 06 年 9 月 H0609	S(发生出错)	
SD2			SD0 的数据以 2 位的 BCD 代码存储更新的日、时。 b15 ~ b8 b7 ~ b0 日 (1~31) 时 (0~23) 例) 25 日 10 时 H2510		
SD3			SD0 的数据以 2 位的 BCD 代码存储更新的分、秒。 b15 ~ b8 b7 ~ b0 分 (0~59) 秒 (0~59) 例) 35 分 48 秒 H3548		
SD4	出错信息分类	出错信息分类 代码	存储用于判断共用信息 (SD5 ~ SD15)、个别信息 (SD16 ~ SD26) 中分别存储的出错信息的分类代码。 b15 ~ b8 b7 ~ b0 个别信息分类代码 共用信息分类代码 - 共用信息分类代码中存储以下代码。 0: 无 1: 模块号 / 基板号 2: 文件名 / 驱动器名 3: 时间 (设置值) 4: 程序出错位置 9: CC-Link Safety 信息 10: 模块号 / 站号 - 个别信息分类代码中存储以下代码。 0: 无 2: 文件名 / 驱动器名 3: 时间 (实测值) 4: 程序出错位置 5: 参数号 6: 报警器 (F) 号 9: 故障信息 10: CC-Link Safety 信息 11: 程序异常结束信息 12: 文件诊断信息	S(发生出错)	QS

附表 10 特殊寄存器

编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU																																													
SD5	出错共用信息	出错共用信息	• 存储出错代码 (SD0) 对应的共用信息。 • 存储的信息有以下 6 种。 1) 模块号 / 基板号 <table><tr><td>编号</td><td colspan="2">内容</td></tr><tr><td>SD5</td><td colspan="2">插槽号/基板号 *1</td></tr><tr><td>SD6</td><td colspan="2">I/O号 *2</td></tr><tr><td>SD7</td><td colspan="2" rowspan="9">(空)</td></tr><tr><td>SD8</td></tr><tr><td>SD9</td></tr><tr><td>SD10</td></tr><tr><td>SD11</td></tr><tr><td>SD12</td></tr><tr><td>SD13</td></tr><tr><td>SD14</td></tr><tr><td>SD15</td></tr></table> *1: SD5(插槽号) 中存储了 255 时, 表示对实际可安装的最终插槽以后的模块执行了指令等。 SD5 中存储基板号时, 将存储 0(主基板)。 *2: SD6(I/O 地址号) 中存储了 FFFF_H 时, 表示可编程控制器参数的 I/O 分配设置中 I/O 地址号的重复等, 不能确定 I/O 编号, 此时应通过 SD5 确定异常位置。 2) 文件名 / 驱动器名 <table><tr><td>编号</td><td colspan="2">内容</td><td rowspan="15">(例) 文件名 = MAIN.QPG b15 ~b8 b7 ~ b0</td></tr><tr><td>SD5</td><td colspan="2">驱动器</td></tr><tr><td>SD6</td><td colspan="2" rowspan="4">文件名 (ASCII码:8字符)</td></tr><tr><td>SD7</td></tr><tr><td>SD8</td></tr><tr><td>SD9</td></tr><tr><td>SD10</td><td>扩展名 *3</td><td>2E_H(.)</td></tr><tr><td>SD11</td><td colspan="2">(ASCII码:3字符)</td></tr><tr><td>SD12</td><td colspan="2" rowspan="4">(空)</td></tr><tr><td>SD13</td></tr><tr><td>SD14</td></tr><tr><td>SD15</td></tr></table>	编号	内容		SD5	插槽号/基板号 *1		SD6	I/O号 *2		SD7	(空)		SD8	SD9	SD10	SD11	SD12	SD13	SD14	SD15	编号	内容		(例) 文件名 = MAIN.QPG b15 ~b8 b7 ~ b0	SD5	驱动器		SD6	文件名 (ASCII码:8字符)		SD7	SD8	SD9	SD10	扩展名 *3	2E _H (.)	SD11	(ASCII码:3字符)		SD12	(空)		SD13	SD14	SD15	S(发生出错)	QS
编号			内容																																															
SD5			插槽号/基板号 *1																																															
SD6			I/O号 *2																																															
SD7			(空)																																															
SD8																																																		
SD9																																																		
SD10																																																		
SD11																																																		
SD12																																																		
SD13																																																		
SD14																																																		
SD15																																																		
编号			内容		(例) 文件名 = MAIN.QPG b15 ~b8 b7 ~ b0																																													
SD5			驱动器																																															
SD6	文件名 (ASCII码:8字符)																																																	
SD7																																																		
SD8																																																		
SD9																																																		
SD10	扩展名 *3	2E _H (.)																																																
SD11	(ASCII码:3字符)																																																	
SD12	(空)																																																	
SD13																																																		
SD14																																																		
SD15																																																		
SD6																																																		
SD7																																																		
SD8																																																		
SD9																																																		
SD10																																																		
SD11																																																		
SD12																																																		
SD13																																																		
SD14																																																		
SD15																																																		

备注

*3 : 扩展名的名称如附表 11 所示。

附表 11 扩展名的名称

SDn	SDn+1		扩展名	文件类型
	高位 8 位	低位 8 位		
51H	50H	41H	QPA	参数
51H	50H	47H	QPG	顺控程序
51H	43H	44H	QCD	软元件注释

附表 10 特殊寄存器

编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU																												
SD5	出错共用信息	出错共用信息	4) 程序出错位置	S(发生出错)	QS																												
			<table><tr><td>编号</td><td>内容</td></tr><tr><td>SD5</td><td rowspan="4">文件名 (ASCII码:8字符)</td></tr><tr><td>SD6</td></tr><tr><td>SD7</td></tr><tr><td>SD8</td></tr><tr><td>SD9</td><td>扩展名 * 3</td><td>2En(.)</td></tr><tr><td>SD10</td><td colspan="2">(ASCII码:3字符)</td></tr><tr><td>SD11</td><td colspan="2">(空)</td></tr><tr><td>SD12</td><td>块号</td><td>* 4</td></tr><tr><td>SD13</td><td>步号</td><td>* 4</td></tr><tr><td>SD14</td><td colspan="2">顺控程序步号(L)</td></tr><tr><td>SD15</td><td colspan="2">顺控程序步号(H)</td></tr></table>			编号	内容	SD5	文件名 (ASCII码:8字符)	SD6	SD7	SD8	SD9	扩展名 * 3	2En(.)	SD10	(ASCII码:3字符)		SD11	(空)		SD12	块号	* 4	SD13	步号	* 4	SD14	顺控程序步号(L)		SD15	顺控程序步号(H)	
编号			内容																														
SD5			文件名 (ASCII码:8字符)																														
SD6																																	
SD7																																	
SD8																																	
SD9			扩展名 * 3			2En(.)																											
SD10			(ASCII码:3字符)																														
SD11			(空)																														
SD12			块号			* 4																											
SD13			步号			* 4																											
SD14			顺控程序步号(L)																														
SD15			顺控程序步号(H)																														
			*4: 块号、步号中被存储 0。																														
SD8	9) CC-Link Safety 信息																																
	<table><tr><td>编号</td><td>内容</td></tr><tr><td>SD5</td><td>出错分类*5</td></tr><tr><td>SD6</td><td>出错项目*5</td></tr><tr><td>SD7</td><td>链接ID</td></tr><tr><td>SD8</td><td>站号</td></tr><tr><td>SD9</td><td>系统区1</td></tr><tr><td>SD10</td><td>系统区2</td></tr><tr><td>SD11</td><td>系统区3</td></tr><tr><td>SD12</td><td>系统区4</td></tr><tr><td>SD13</td><td>系统区5</td></tr><tr><td>SD14</td><td>系统区6</td></tr><tr><td>SD15</td><td>系统区7</td></tr><tr><td>SD16</td><td>系统区8</td></tr></table>	编号	内容	SD5	出错分类*5	SD6	出错项目*5	SD7	链接ID	SD8	站号	SD9	系统区1	SD10	系统区2	SD11	系统区3	SD12	系统区4	SD13	系统区5	SD14	系统区6	SD15	系统区7	SD16	系统区8						
编号	内容																																
SD5	出错分类*5																																
SD6	出错项目*5																																
SD7	链接ID																																
SD8	站号																																
SD9	系统区1																																
SD10	系统区2																																
SD11	系统区3																																
SD12	系统区4																																
SD13	系统区5																																
SD14	系统区6																																
SD15	系统区7																																
SD16	系统区8																																
SD9																																	
SD10																																	
	*5: 只有在出错代码为 8300(CC-LINK REMOTE DETECTION ERROR) 时才存储出错分类、出错项目。 出错代码为 8300 以外时存储 0。																																
SD11	10) 模块号 / 站号																																
SD12	<table><tr><td>编号</td><td>内容</td></tr><tr><td>SD5</td><td>插槽号</td></tr><tr><td>SD6</td><td>I/O号</td></tr><tr><td>SD7</td><td>站号</td></tr><tr><td>SD8</td><td rowspan="8">(空)</td></tr><tr><td>SD9</td></tr><tr><td>SD10</td></tr><tr><td>SD11</td></tr><tr><td>SD12</td></tr><tr><td>SD13</td></tr><tr><td>SD14</td></tr><tr><td>SD15</td></tr></table>	编号	内容	SD5	插槽号	SD6	I/O号	SD7	站号	SD8	(空)	SD9	SD10	SD11	SD12	SD13	SD14	SD15															
编号	内容																																
SD5	插槽号																																
SD6	I/O号																																
SD7	站号																																
SD8	(空)																																
SD9																																	
SD10																																	
SD11																																	
SD12																																	
SD13																																	
SD14																																	
SD15																																	
SD13																																	
SD14																																	
SD15																																	

附表 10 特殊寄存器

编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU																										
SD16	出错个别信息	出错个别信息	- 出错代码 (SD0) 中存储对应的个别信息。 - 存储的信息有以下 9 种。	S(发生出错)	QS																										
SD17			2) 文件名 / 驱动器名																												
SD18			<table><tr><td>编号</td><td>内容</td><td rowspan="12">(例) 文件名=MAIN.QPG b15 ~ b8 b7 ~ b0</td></tr><tr><td>SD16</td><td>驱动器</td></tr><tr><td>SD17</td><td rowspan="3">文件名 (ASCII码:8字符)</td><td>41h(A) 40h(M)</td></tr><tr><td>SD18</td><td>43h(N) 49h(I)</td></tr><tr><td>SD19</td><td>20h(SP) 20x(SP)</td></tr><tr><td>SD20</td><td rowspan="2">扩展名*3 2Eh(.) (ASCII码:3字符)</td><td>20h(SP) 20h(SP)</td></tr><tr><td>SD21</td><td>51h(Q) 2Eh(.)</td></tr><tr><td>SD22</td><td rowspan="5">(空)</td><td>47h(G) 50h(P)</td></tr><tr><td>SD23</td></tr><tr><td>SD24</td></tr><tr><td>SD25</td></tr><tr><td>SD26</td></tr></table>			编号	内容	(例) 文件名=MAIN.QPG b15 ~ b8 b7 ~ b0	SD16	驱动器	SD17	文件名 (ASCII码:8字符)	41h(A) 40h(M)	SD18	43h(N) 49h(I)	SD19	20h(SP) 20x(SP)	SD20	扩展名*3 2Eh(.) (ASCII码:3字符)	20h(SP) 20h(SP)	SD21	51h(Q) 2Eh(.)	SD22	(空)	47h(G) 50h(P)	SD23	SD24	SD25	SD26		
编号			内容			(例) 文件名=MAIN.QPG b15 ~ b8 b7 ~ b0																									
SD16			驱动器																												
SD17			文件名 (ASCII码:8字符)				41h(A) 40h(M)																								
SD18							43h(N) 49h(I)																								
SD19							20h(SP) 20x(SP)																								
SD20			扩展名*3 2Eh(.) (ASCII码:3字符)				20h(SP) 20h(SP)																								
SD21							51h(Q) 2Eh(.)																								
SD22			(空)				47h(G) 50h(P)																								
SD23																															
SD24																															
SD25																															
SD26																															
SD19			3) 时间 (实测值)																												
SD20			<table><tr><td>编号</td><td>内容</td></tr><tr><td>SD16</td><td>时间:单位1μs(0~999μs)</td></tr><tr><td>SD17</td><td>时间:单位1ms(0~65535ms)</td></tr><tr><td>SD18</td><td rowspan="9">(空)</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td></tr><tr><td>SD22</td></tr><tr><td>SD23</td></tr><tr><td>SD24</td></tr><tr><td>SD25</td></tr><tr><td>SD26</td></tr></table>			编号	内容	SD16	时间:单位1μs(0~999μs)	SD17	时间:单位1ms(0~65535ms)	SD18	(空)	SD19	SD20	SD21	SD22	SD23	SD24	SD25	SD26										
编号			内容																												
SD16			时间:单位1μs(0~999μs)																												
SD17			时间:单位1ms(0~65535ms)																												
SD18			(空)																												
SD19																															
SD20																															
SD21																															
SD22																															
SD23																															
SD24																															
SD25																															
SD26																															
SD21	4) 程序出错位置																														
SD22	<table><tr><td>编号</td><td>内容</td></tr><tr><td>SD16</td><td rowspan="3">文件名 (ASCII码:8字符)</td></tr><tr><td>SD17</td></tr><tr><td>SD18</td></tr><tr><td>SD19</td><td rowspan="2">扩展名*3 2Eh(.) (ASCII码:3字符)</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td><td>(空)</td></tr><tr><td>SD22</td><td>块号*6</td></tr><tr><td>SD23</td><td>步号*6</td></tr><tr><td>SD24</td><td>顺控程序步号(L)</td></tr><tr><td>SD25</td><td>顺控程序步号(H)</td></tr><tr><td>SD26</td><td></td></tr></table>	编号	内容	SD16	文件名 (ASCII码:8字符)	SD17	SD18	SD19	扩展名*3 2Eh(.) (ASCII码:3字符)	SD20	SD21	(空)	SD22	块号*6	SD23	步号*6	SD24	顺控程序步号(L)	SD25	顺控程序步号(H)	SD26										
编号	内容																														
SD16	文件名 (ASCII码:8字符)																														
SD17																															
SD18																															
SD19	扩展名*3 2Eh(.) (ASCII码:3字符)																														
SD20																															
SD21	(空)																														
SD22	块号*6																														
SD23	步号*6																														
SD24	顺控程序步号(L)																														
SD25	顺控程序步号(H)																														
SD26																															
SD23	*6: 块号、步号中被存储 0。																														
SD24	5) 参数号 6) 报警器号																														
SD25	<table><tr><td>编号</td><td>内容</td><td>编号</td><td>内容</td></tr><tr><td>SD16</td><td>参数号</td><td>SD16</td><td>No.</td></tr><tr><td>SD17</td><td rowspan="10">(空)</td><td>SD17</td><td rowspan="10">(空)</td></tr><tr><td>SD18</td><td>SD18</td></tr><tr><td>SD19</td><td>SD19</td></tr><tr><td>SD20</td><td>SD20</td></tr><tr><td>SD21</td><td>SD21</td></tr><tr><td>SD22</td><td>SD22</td></tr><tr><td>SD23</td><td>SD23</td></tr><tr><td>SD24</td><td>SD24</td></tr><tr><td>SD25</td><td>SD25</td></tr><tr><td>SD26</td><td>SD26</td></tr></table>	编号	内容	编号	内容	SD16	参数号	SD16	No.	SD17	(空)	SD17	(空)	SD18	SD18	SD19	SD19	SD20	SD20	SD21	SD21	SD22	SD22	SD23	SD23	SD24	SD24	SD25	SD25	SD26	SD26
编号	内容	编号	内容																												
SD16	参数号	SD16	No.																												
SD17	(空)	SD17	(空)																												
SD18		SD18																													
SD19		SD19																													
SD20		SD20																													
SD21		SD21																													
SD22		SD22																													
SD23		SD23																													
SD24		SD24																													
SD25		SD25																													
SD26		SD26																													
SD26	9) 故障信息																														
	<table><tr><td>编号</td><td>内容</td></tr><tr><td>SD16</td><td>故障信息1</td></tr><tr><td>SD17</td><td>故障信息2</td></tr><tr><td>SD18</td><td>故障信息3</td></tr><tr><td>SD19</td><td>故障信息4</td></tr><tr><td>SD20</td><td>故障信息5</td></tr><tr><td>SD21</td><td>故障信息6</td></tr><tr><td>SD22</td><td>故障信息7</td></tr><tr><td>SD23</td><td>故障信息8</td></tr><tr><td>SD24</td><td>故障信息9</td></tr><tr><td>SD25</td><td>故障信息10</td></tr><tr><td>SD26</td><td>故障信息11</td></tr></table>	编号	内容	SD16	故障信息1	SD17	故障信息2	SD18	故障信息3	SD19	故障信息4	SD20	故障信息5	SD21	故障信息6	SD22	故障信息7	SD23	故障信息8	SD24	故障信息9	SD25	故障信息10	SD26	故障信息11						
编号	内容																														
SD16	故障信息1																														
SD17	故障信息2																														
SD18	故障信息3																														
SD19	故障信息4																														
SD20	故障信息5																														
SD21	故障信息6																														
SD22	故障信息7																														
SD23	故障信息8																														
SD24	故障信息9																														
SD25	故障信息10																														
SD26	故障信息11																														

附表 10 特殊寄存器

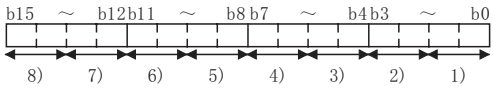
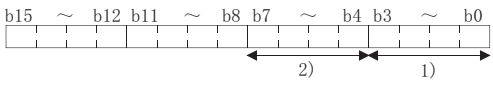
编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU																								
SD16	出错个别信息	出错个别信息	10) CC-Link Safety 信息	S(发生出错)	QS																								
SD17			<table><tr><th>编号</th><th>内容</th></tr><tr><td>SD16</td><td>个别信息项目数</td></tr><tr><td>SD17</td><td>个别信息1</td></tr><tr><td>SD18</td><td>个别信息2</td></tr><tr><td>SD19</td><td>个别信息3</td></tr><tr><td>SD20</td><td>个别信息4</td></tr><tr><td>SD21</td><td>个别信息5</td></tr><tr><td>SD22</td><td>个别信息6</td></tr><tr><td>SD23</td><td>个别信息7</td></tr><tr><td>SD24</td><td>个别信息8</td></tr><tr><td>SD25</td><td>个别信息9</td></tr><tr><td>SD26</td><td>个别信息10</td></tr></table>			编号	内容	SD16	个别信息项目数	SD17	个别信息1	SD18	个别信息2	SD19	个别信息3	SD20	个别信息4	SD21	个别信息5	SD22	个别信息6	SD23	个别信息7	SD24	个别信息8	SD25	个别信息9	SD26	个别信息10
编号			内容																										
SD16			个别信息项目数																										
SD17			个别信息1																										
SD18			个别信息2																										
SD19			个别信息3																										
SD20			个别信息4																										
SD21			个别信息5																										
SD22			个别信息6																										
SD23	个别信息7																												
SD24	个别信息8																												
SD25	个别信息9																												
SD26	个别信息10																												
SD18																													
SD19																													
SD20																													
SD21																													
SD22																													
SD23																													
SD24																													
SD25																													
SD26																													
SD27	诊断出错发生源 CPU 的识别符	CPU 识别符 (CPU A/CPU B)	• 存储发生了 SD0 ~ SD26 的诊断出错的 CPU 的 CPU 识别符。 0001H : CPU A 0002H : CPU B	S(发生出错)																									
SD50	出错解除	出错解除的出错代码	• 存储出错解除的出错代码。	U																									
SD51	电池电量不足锁存	发生电池电量不足的对象位模式	• 发生了电池电量不足时，对应的位将变为 1(ON)。 • 以后，即使电池电压恢复正常也保持 1 不变。 <table><tr><td>b15</td><td>~</td><td>b1</td><td>b0</td></tr><tr><td colspan="4">0</td></tr></table> CPU模块用电池出错	b15	~	b1	b0	0				S(发生出错)																	
b15	~	b1	b0																										
0																													
SD52	电池电量不足	发生电池电量不足的对象位模式	• 与上述 SD51 的构成相同。 • 以后，电池电压恢复正常时将变为 0(OFF)。	S(发生出错)																									
SD53	检测出 AC DOWN	检测出 AC DOWN 的次数	• 每次CPU模块的运算过程中输入电压为额定的85%(AC电源)以下时将被 +1，将以 BIN 代码存储值。	S(发生出错)																									

附表 10 特殊寄存器

编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU																																																																			
SD61	I/O 模块校验 出错号	I/O 模块校验出 错模块号	• 存储发生了 I/O 模块校验出错的模块的最小号的 I/O 地址号。	S(发生出错)	QS																																																																			
SD62	报警器编号	报警器编号	• 存储最先检测出的报警器的编号 (F 编号)。	S(执行指令)																																																																				
SD63	报警器个数	报警器个数	• 存储检测出的报警器的个数。	S(执行指令)																																																																				
SD64	报警器检测编 号表	报警器检测编号	• 通过 [OUT F] 、 [SET F] 使 F 为 ON 时, SD64～SD79 中依次 ON 的 F 编号将被登录。	S(执行指令)																																																																				
SD65			• 通过 [RST F] 将已 OFF 的 F 编号从 SD64～SD79 中删除, 存储在被删除的 F 编号以后的 F 编号将向前对齐。																																																																					
SD66			检测出的报警器个数有 16 个时, 即使检测出了第 17 个, 也不会被存储到 SD64～SD79 中。																																																																					
SD67																																																																								
SD68																																																																								
SD69			SET SET SET RST SET SET SET SET SET SET SET SET F50 F25 F99 F25 F15 F70 F65 F38 F110 F151 F210 F50																																																																					
SD70			SD62 <table><tr><td>0</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>99</td></tr></table> (检测出的编号)			0	50	50	50	50	50	50	50	50	50	50	50	99																																																						
0			50			50	50	50	50	50	50	50	50	50	50	99																																																								
SD71			SD63 <table><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>8</td></tr></table> (检测出的个数)			0	1	2	3	2	3	4	5	6	7	8	9	8																																																						
0			1			2	3	2	3	4	5	6	7	8	9	8																																																								
SD72			SD64 <table><tr><td>0</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>99</td></tr></table>			0	50	50	50	50	50	50	50	50	50	50	50	99																																																						
0			50			50	50	50	50	50	50	50	50	50	50	99																																																								
SD73	SD65 <table><tr><td>0</td><td>0</td><td>25</td><td>25</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>15</td></tr></table>	0	0	25		25	99	99	99	99	99	99	99	99	15																																																									
0	0	25	25	99	99	99	99	99	99	99	99	15																																																												
SD74	SD66 <table><tr><td>0</td><td>0</td><td>0</td><td>99</td><td>0</td><td>15</td><td>15</td><td>15</td><td>15</td><td>15</td><td>15</td><td>15</td><td>70</td></tr></table>	0	0	0	99	0	15	15	15	15	15	15	15	70																																																										
0	0	0	99	0	15	15	15	15	15	15	15	70																																																												
SD75	SD67 <table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>70</td><td>70</td><td>70</td><td>70</td><td>70</td><td>70</td><td>65</td></tr></table>	0	0	0	0	0	0	70	70	70	70	70	70	65																																																										
0	0	0	0	0	0	70	70	70	70	70	70	65																																																												
SD76	SD68 <table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>65</td><td>65</td><td>65</td><td>65</td><td>65</td><td>38</td></tr></table>	0	0	0	0	0	0	0	65	65	65	65	65	38																																																										
0	0	0	0	0	0	0	65	65	65	65	65	38																																																												
SD77	SD69 <table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>38</td><td>38</td><td>38</td><td>38</td><td>110</td></tr></table>	0	0	0	0	0	0	0	0	38	38	38	38	110																																																										
0	0	0	0	0	0	0	0	38	38	38	38	110																																																												
SD78	SD70 <table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>110</td><td>110</td><td>110</td><td>151</td></tr></table>	0	0	0	0	0	0	0	0	0	110	110	110	151																																																										
0	0	0	0	0	0	0	0	0	110	110	110	151																																																												
SD79	SD71 <table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>151</td><td>151</td><td>210</td></tr></table>	0	0	0	0	0	0	0	0	0	0	151	151	210																																																										
0	0	0	0	0	0	0	0	0	0	151	151	210																																																												
			SD72 <table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>210</td><td>0</td></tr></table>	0	0	0	0	0	0	0	0	0	0	0	210	0																																																								
0	0	0	0	0	0	0	0	0	0	0	210	0																																																												
			SD73 <table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0	0	0	0	0	0	0	0																																																								
0	0	0	0	0	0	0	0	0	0	0	0	0																																																												
			SD74 <table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0	0	0	0	0	0	0	0																																																								
0	0	0	0	0	0	0	0	0	0	0	0	0																																																												
			SD75 <table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0	0	0	0	0	0	0	0																																																								
0	0	0	0	0	0	0	0	0	0	0	0	0																																																												
			SD76 <table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0	0	0	0	0	0	0	0																																																								
0	0	0	0	0	0	0	0	0	0	0	0	0																																																												
			SD77 <table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0	0	0	0	0	0	0	0																																																								
0	0	0	0	0	0	0	0	0	0	0	0	0																																																												
			SD78 <table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0	0	0	0	0	0	0	0																																																								
0	0	0	0	0	0	0	0	0	0	0	0	0																																																												
			SD79 <table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0	0	0	0	0	0	0	0																																																								
0	0	0	0	0	0	0	0	0	0	0	0	0																																																												
				(检测出的编号)																																																																				
SD81	出错原因	出错原因	• 发生了继续运行出错时, 对应的位将全部 ON。 • 消除了原因后, 进行了出错解除、可编程控制器的电源重启或者 CPU 模块的复位解除时该报警器将 OFF。 <table><tr><th>位号</th><th>原因名称</th></tr><tr><td>0</td><td>发生了瞬时停电</td></tr><tr><td>1</td><td>电池电量不足</td></tr><tr><td>2</td><td>内置 ROM 写入次数溢出</td></tr><tr><td>3</td><td>测试模式下连续 RUN 允许时间溢出</td></tr><tr><td>4</td><td>扫描时间溢出</td></tr><tr><td>5</td><td>报警器 ON</td></tr><tr><td>6</td><td>安全远程站检测出出错</td></tr><tr><td>7</td><td>安全远程站产品信息不一致</td></tr><tr><td>8</td><td>初始化监视超时出错 安全监视超时出错 出错监视超时出错</td></tr><tr><td>9</td><td>安全远程站指令异常 安全远程站数据分割异常 安全远程站链接 ID 异常 安全远程站运行号异常 安全远程站接收数据异常</td></tr><tr><td>10—15</td><td>(空)</td></tr></table>	位号	原因名称	0	发生了瞬时停电	1	电池电量不足	2	内置 ROM 写入次数溢出	3	测试模式下连续 RUN 允许时间溢出	4	扫描时间溢出	5	报警器 ON	6	安全远程站检测出出错	7	安全远程站产品信息不一致	8	初始化监视超时出错 安全监视超时出错 出错监视超时出错	9	安全远程站指令异常 安全远程站数据分割异常 安全远程站链接 ID 异常 安全远程站运行号异常 安全远程站接收数据异常	10—15	(空)	S(发生出错)																																												
位号	原因名称																																																																							
0	发生了瞬时停电																																																																							
1	电池电量不足																																																																							
2	内置 ROM 写入次数溢出																																																																							
3	测试模式下连续 RUN 允许时间溢出																																																																							
4	扫描时间溢出																																																																							
5	报警器 ON																																																																							
6	安全远程站检测出出错																																																																							
7	安全远程站产品信息不一致																																																																							
8	初始化监视超时出错 安全监视超时出错 出错监视超时出错																																																																							
9	安全远程站指令异常 安全远程站数据分割异常 安全远程站链接 ID 异常 安全远程站运行号异常 安全远程站接收数据异常																																																																							
10—15	(空)																																																																							
SD150	I/O 模块校验 出错	校验出错模块的 16 点单位的位 模式 0: 无 I/O 校验 出错 1: 有 I/O 校验 出错	• 电源 ON 时检测到与登录的 I/O 模块信息不相同的 I/O 模块时, 存储该 I/O 模块编号 (16 点单位)。 (参数设置时设置的 I/O 模块号) <table><tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr><tr><td>SD150</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1 (1₁₆)</td></tr><tr><td>SD151</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1 (1₁₆)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>SD153</td><td>0</td><td>1 (1₁₆)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> ↑ 显示 I/O 模块校验出错		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	SD150	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1 (1 ₁₆)	SD151	0	0	0	0	0	0	1 (1 ₁₆)	0	0	0	0	0	0	0	0	0	SD153	0	1 (1 ₁₆)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	S(发生出错)
			b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																						
SD150			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1 (1 ₁₆)																																																						
SD151			0	0	0	0	0	0	1 (1 ₁₆)	0	0	0	0	0	0	0	0	0																																																						
SD153	0	1 (1 ₁₆)	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																																								
SD151																																																																								
SD152																																																																								
SD153	• 即使恢复正常也不被清除。 如果进行出错解除则将被清除。																																																																							

(2) 系统信息

附表 12 特殊寄存器

编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU
SD200	开关状态	CPU 开关状态	• 通过以下格式存储 CPU 模块的开关状态。  1): CPU 开关状态 0: RUN 1: STOP 2: RESET	S (每次 END)	
SD201	LED 状态	CPU-LED 状态	• CPU 模块的 LED 的状态按以下的位模式被存储。 • 0 表示熄灯, 1 表示亮灯, 2 表示闪烁。  1): RUN 5): 空 2): ERR. 6): 空 3): USER 7): TEST 4): BAT. 8): 空	S (状态变化)	QS
SD203	CPU 动作状态	CPU 动作状态	• 按下图所示存储 CPU 模块的动作状态。  1): CPU 动作状态 0: RUN 2: STOP 2): STOP 原因 0: RUN/STOP/RESET 开关 1: 远程触点 2: 通过 GX Developer 进行的远程操作 4: 出错 5: 安全模式 (再启动等待) 6: 可编程控制器写入执行完毕 注) 上述的数字中较小的数字的原因将被优先存储。 但是, “4: 出错” 将被作为最高优先顺序处理。	S (每次 END)	

附表 12 特殊寄存器

编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU
SD210	时钟数据	时钟数据 (公历, 月)	• SD210 中按下图所示以 BCD 代码存储年 (公历, 低 2 位)、月。 b15 ~ b12 b11 ~ b8 b7 ~ b4 b3 ~ b0 例) 2006年, 9月 0609H 年月	S(请求时)	QS
SD211	时钟数据	时钟数据 (日, 时)	• SD211 中按下图所示以 BCD 代码存储日、时。 b15 ~ b12 b11 ~ b8 b7 ~ b4 b3 ~ b0 例) 25日, 10时 2510H 日时		
SD212	时钟数据	时钟数据 (分, 秒)	• SD212 中按下图所示以 BCD 代码存储分、秒。 b15 ~ b12 b11 ~ b8 b7 ~ b4 b3 ~ b0 例) 35分, 48秒 3548H 分秒		
SD213	时钟数据	时钟数据 (公历高位, 星期)	• SD213 中按下图所示以 BCD 代码存储年 (公历, 高 2 位) 及星期。 b15 ~ b12 b11 ~ b8 b7 ~ b4 b3 ~ b0 例) 2006年, 星期一 2001H 公历高位 (0~99) 星期 0 日 1 一 2 二 3 三 4 四 5 五 6 六		
SD232	ROM 写入次数	目前为止的 ROM 写入次数	• 存储目前为止的 ROM 写入次数。	S(ROM 化时)	S(初始化)
SD233					
SD240	基本模式	0: 自动模式	• 存储基本模式。(固定为 0)		
SD241	扩展级数	0: 仅主基板	• 存储实际安装的扩展基板的最多级数。(固定为 0)	S(初始化)	
SD242	Q 基板安装有无	基板类型的判别 0: 未安装基板 1: 安装了 QS***B	b15 ~ b1 b0 空主基板	S(初始化)	
SD243	基板插槽个数 (动作状态)	基板插槽个数	b15 ~ b4 b3 ~ b0 SD243 空 主基板 SD244 空 • 在上述各区域中, 存储实际安装的基板的插槽数。 (参数设置时设置的插槽个数)	S(初始化)	
SD244					
SD245	基板插槽个数 (安装状态)	基板插槽个数	b15 ~ b4 b3 ~ b0 SD245 空 主基板 SD246 空 • 在上述各区域中存储安装基板上的插槽数。(安装基板的实际插槽数)	S(初始化)	
SD246					
SD250	实际安装最大 I/O	实际安装最大 I/O 地址号	• 以 BIN 值存储实际安装的模块的最后 I/O 地址号 +1 的高位 2 位。	S(初始化)	

附表 12 特殊寄存器

编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU
SD254	MELSECNET/H 信息	安装个数	• 显示安装的 MELSECNET/H 模块的个数。	S(初始化)	QS
SD255		I/O 地址号	• 显示安装的 MELSECNET/H 模块的 I/O 地址号。		
SD256		网络号	• 显示安装的 MELSECNET/H 模块的网络号。		
SD257		组号	• 显示安装的 MELSECNET/H 模块的组号。		
SD258		站号	• 显示安装的 MELSECNET/H 模块的站号。		
SD290	软元件分配 (与参数的内容 相同)	X 分配点数	• 存储当前设置的软元件 X 的点数。	S(初始化)	
SD291		Y 分配点数	• 存储当前设置的软元件 Y 的点数。		
SD292		M 分配点数	• 存储当前设置的软元件 M 的点数。		
SD294		B 分配点数	• 存储当前设置的软元件 B 的点数。		
SD295		F 分配点数	• 存储当前设置的软元件 F 的点数。		
SD296		SB 分配点数	• 存储当前设置的软元件 SB 的点数。		
SD297		V 分配点数	• 存储当前设置的软元件 V 的点数。		
SD299		T 分配点数	• 存储当前设置的软元件 T 的点数。		
SD300		ST 分配点数	• 存储当前设置的软元件 ST 的点数。		
SD301		C 分配点数	• 存储当前设置的软元件 C 的点数。		
SD302		D 分配点数	• 存储当前设置的软元件 D 的点数。		
SD303		W 分配点数	• 存储当前设置的软元件 W 的点数。		
SD304		SW 分配点数	• 存储当前设置的软元件 SW 的点数。		

(3) 系统时钟 / 计数器

附表 13 特殊寄存器

编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU
SD412	1 秒计数器	以 1 秒为单位的计数	- CPU 模块投运后, 每隔 1 秒 +1。 - 计数在 0 → 32767 → -32768 → 0 的范围重复。	S(状态变化)	QS
SD414	2n 秒时钟设置	2n 秒时钟的单位	- 存储 2n 秒时钟的 n。(缺省 :30) - 设置范围为 1 ~ 32767。	U	
SD420	扫描计数器	每 1 次扫描的计数	- CPU 模块投运后, 每 1 次扫描 +1。 - 计数在 0 → 32767 → -32768 → 0 的范围重复。	S(每次 END)	

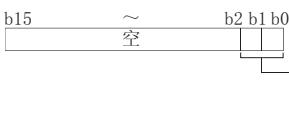
(4) 扫描信息

附表 14 特殊寄存器

编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU
SD520	当前扫描时间	当前扫描时间 (单位 :ms)	• 在 SD520、SD521 中存储当前的扫描时间。 (计测以 100 μ s 为单位进行。) SD520: 存储 ms 的位 (存储范围 :0 ~ 6553) SD521: 存储 μ s 的位 (存储范围 :0 ~ 900) (例) 当前的扫描时间为 23.6ms 时, 按以下方式存储。 SD520 = 23 SD521 = 600 • 扫描时间的精度为 ± 0.1 ms。	S(每次 END)	QS
SD521		当前扫描时间 (单位 : μ s)			
SD524	最小扫描时间	最小扫描时间 (单位 :ms)	• 在 SD524、SD525 中存储扫描时间的最小值。 (计测以 100 μ s 为单位进行。) SD524: 存储 ms 的位 (存储范围 :0 ~ 6553) SD525: 存储 μ s 的位 (存储范围 :0 ~ 900) • 扫描时间的精度为 ± 0.1 ms。	S(每次 END)	
SD525		最小扫描时间 (单位 : μ s)			
SD526	最大扫描时间	最大扫描时间 (单位 :ms)	• 在 SD526、SD527 中存储扫描时间的最大值。 (计测以 100 μ s 为单位进行) SD526: 存储 ms 的位 (存储范围 :0 ~ 6553) SD527: 存储 μ s 的位 (存储范围 :0 ~ 900) • 扫描时间的精度为 ± 0.1 ms。	S(每次 END)	
SD527		最大扫描时间 (单位 : μ s)			
SD540	END 处理时间	END 处理时间 (单位 :ms)	• 将扫描程序结束后, 至下一次扫描开始为止的时间存储到 SD540、SD541 中。 (计测以 100 μ s 为单位进行。 .) SD540: 存储 ms 的位 (存储范围 :0 ~ 6553) SD541: 存储 μ s 的位 (存储范围 :0 ~ 900) • END 处理时间的精度为 ± 0.1 ms。	S(每次 END)	
SD541		END 处理时间 (单位 : μ s)			
SD542	恒定扫描等待时间	恒定扫描等待时间 (单位 :ms)	• 将恒定扫描设置时的等待时间存储到 SD542、SD543 中。 (计测以 100 μ s 为单位进行。) SD542: 存储 ms 的位 (存储范围 :0 ~ 6553) SD543: 存储 μ s 的位 (存储范围 :0 ~ 900) • 恒定扫描等待时间的精度为 ± 0.1 ms。	S(每次 END)	
SD543		恒定扫描等待时间 (单位 : μ s)			
SD548	扫描程序执行时间	扫描程序执行时间 (单位 :ms)	• 将 1 次扫描中的扫描程序的执行时间存储到 SD548、SD549 中。 (计测以 100 μ s 为单位进行。) SD548: 存储 ms 的位 (存储范围 :0 ~ 6553) SD549: 存储 μ s 的位 (存储范围 :0 ~ 900) • 每次扫描存储。 • 扫描程序执行时间的精度为 ± 0.1 ms。	S(每次 END)	
SD549		扫描程序执行时间 (单位 : μ s)			

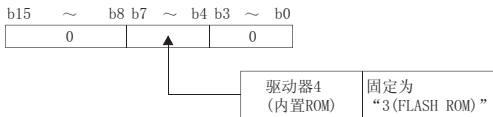
(5) 安全 CPU

附表 15 特殊寄存器

编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU
SD560	安全 CPU 动作模式	安全 CPU 动作模式	存储安全 CPU 动作模式。  00 : 安全模式 01 : 测试模式 10 : 安全模式 (再启动等待)	S(状态变化)	QS
SD561	测试模式连续 RUN 时间	测试模式连续 RUN 时间 (秒)	- 存储测试模式下的连续 RUN 时间。(以 1 秒为单位计测。) (测试模式下的运行时间。在 STOP → RUN 的时点开始计测。) (不包含 STOP 的时间。)	S(每次 END)	
SD562			计测值的存储范围为 :1 ~ 2147483647。		

(6) 存储器

附表 16 特殊寄存器

编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU
SD620	存储器类型	存储器类型	显示内置的存储器的类型。 	S(初始化)	QS
SD623	驱动器 4 (ROM) 容量	驱动器4的容量	以 1k 字节为单位存储驱动器 4 的容量。	S(初始化)	

(7) CC-Link Safety

附表 17 特殊寄存器

编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU																														
SD1000 ~ SD1003	安全远程站指定 (第 1 个 CC-Link Safety 主站模块)	0: 无安全远程站指定 1: 有安全远程站指定	• 存储安全远程站的指定状态。 • 标准远程站时存储 “0”。 <table border="1"> <tr> <td></td><td>b15</td><td>b14</td><td>-</td><td>b1</td><td>b0</td></tr> <tr> <td>SD1000</td><td>16</td><td>15</td><td>~</td><td>2</td><td>1</td></tr> <tr> <td>SD1001</td><td>32</td><td>31</td><td>~</td><td>18</td><td>17</td></tr> <tr> <td>SD1002</td><td>48</td><td>47</td><td>~</td><td>34</td><td>33</td></tr> <tr> <td>SD1003</td><td>64</td><td>63</td><td>~</td><td>50</td><td>49</td></tr> </table> 表中 1 ~ 64 表示站号		b15	b14	-	b1	b0	SD1000	16	15	~	2	1	SD1001	32	31	~	18	17	SD1002	48	47	~	34	33	SD1003	64	63	~	50	49	S(初始化)	
	b15	b14	-	b1	b0																														
SD1000	16	15	~	2	1																														
SD1001	32	31	~	18	17																														
SD1002	48	47	~	34	33																														
SD1003	64	63	~	50	49																														
SD1004 ~ SD1007	安全站刷新通信状态 (第 1 个 CC-Link Safety 主站模块)	0: 正常, 预约站指定, 未使用, 标准远程站 1: 安全站通信异常	• 存储安全远程站的刷新通信状态。 • 标准远程站时存储 “0”。 <table border="1"> <tr> <td></td><td>b15</td><td>b14</td><td>-</td><td>b1</td><td>b0</td></tr> <tr> <td>SD1004</td><td>16</td><td>15</td><td>~</td><td>2</td><td>1</td></tr> <tr> <td>SD1005</td><td>32</td><td>31</td><td>~</td><td>18</td><td>17</td></tr> <tr> <td>SD1006</td><td>48</td><td>47</td><td>~</td><td>34</td><td>33</td></tr> <tr> <td>SD1007</td><td>64</td><td>63</td><td>~</td><td>50</td><td>49</td></tr> </table> 表中 1 ~ 64 表示站号		b15	b14	-	b1	b0	SD1004	16	15	~	2	1	SD1005	32	31	~	18	17	SD1006	48	47	~	34	33	SD1007	64	63	~	50	49	S(状态变化)	
	b15	b14	-	b1	b0																														
SD1004	16	15	~	2	1																														
SD1005	32	31	~	18	17																														
SD1006	48	47	~	34	33																														
SD1007	64	63	~	50	49																														
SD1008 ~ SD1071	安全站通信状态 (第 1 个 CC-Link Safety 主站模块)	存储与安全远程站的通信状态	• 存储与各安全远程站的通信状态。 • SD1008: 站号 1 ~ SD1071: 站号 64 • (标准远程站、预约站指定、未连接时固定为 0) 0: 正在进行正常通信 10: 正在进行初始化 20: 正在存取内部信息 30: 链接异常 8300: 链接异常 (安全远程站检测出出错) 8310: 链接异常 (产品信息不一致) 8320: 链接异常 (初始化监视超时) 8321: 链接异常 (安全监视超时) 8322: 链接异常 (出错监视超时) 8330: 链接异常 (指令异常) 8331: 链接异常 (数据分割号异常) 8332: 链接异常 (链接 ID 异常) 8333: 链接异常 (运行号异常) 8334: 链接异常 (接收数据异常)	S(状态变化)	Q ^S																														
SD1072 ~ SD1075	安全站互锁状态 (第 1 个 CC-Link Safety 主站模块)	0: 未互锁 1: 互锁中	如果在互锁状态下主站中检测出出错, 相应站号对应的位将变为 1。 <table border="1"> <tr> <td></td><td>b15</td><td>b14</td><td>-</td><td>b1</td><td>b0</td></tr> <tr> <td>SD1072</td><td>16</td><td>15</td><td>~</td><td>2</td><td>1</td></tr> <tr> <td>SD1073</td><td>32</td><td>31</td><td>~</td><td>18</td><td>17</td></tr> <tr> <td>SD1074</td><td>48</td><td>47</td><td>~</td><td>34</td><td>33</td></tr> <tr> <td>SD1075</td><td>64</td><td>63</td><td>~</td><td>50</td><td>49</td></tr> </table> 表中 1 ~ 64 表示站号		b15	b14	-	b1	b0	SD1072	16	15	~	2	1	SD1073	32	31	~	18	17	SD1074	48	47	~	34	33	SD1075	64	63	~	50	49	S(状态变化)	
	b15	b14	-	b1	b0																														
SD1072	16	15	~	2	1																														
SD1073	32	31	~	18	17																														
SD1074	48	47	~	34	33																														
SD1075	64	63	~	50	49																														
SD1076 ~ SD1079	安全站互锁解除请求 (第 1 个 CC-Link Safety 主站模块)	0: 不解除安全站的 I/O 互锁 1: 解除安全站的 I/O 互锁	通过将本寄存器的位由 0 变为 1, 解除安全站的 I/O 互锁。 <table border="1"> <tr> <td></td><td>b15</td><td>b14</td><td>-</td><td>b1</td><td>b0</td></tr> <tr> <td>SD1076</td><td>16</td><td>15</td><td>~</td><td>2</td><td>1</td></tr> <tr> <td>SD1077</td><td>32</td><td>31</td><td>~</td><td>18</td><td>17</td></tr> <tr> <td>SD1078</td><td>48</td><td>47</td><td>~</td><td>34</td><td>33</td></tr> <tr> <td>SD1079</td><td>64</td><td>63</td><td>~</td><td>50</td><td>49</td></tr> </table> 表中 1 ~ 64 表示站号		b15	b14	-	b1	b0	SD1076	16	15	~	2	1	SD1077	32	31	~	18	17	SD1078	48	47	~	34	33	SD1079	64	63	~	50	49	U(请求时)	
	b15	b14	-	b1	b0																														
SD1076	16	15	~	2	1																														
SD1077	32	31	~	18	17																														
SD1078	48	47	~	34	33																														
SD1079	64	63	~	50	49																														

附表 17 特殊寄存器

编号	名称	内容	详细内容	设置侧 (设置时间)	对应 CPU																														
SD1200 ～ SD1203	安全远程站指定 (第 2 个 CC-Link Safety 主站模块)	0: 无安全远程站指定 1: 有安全远程站指定	- 存储安全远程站的指定状态。 - 标准远程站时存储“0”。 <table border="1"> <tr> <td></td><td>b15</td><td>b14</td><td>—</td><td>b1</td><td>b0</td></tr> <tr> <td>SD1000</td><td>16</td><td>15</td><td>～</td><td>2</td><td>1</td></tr> <tr> <td>SD1001</td><td>32</td><td>31</td><td>～</td><td>18</td><td>17</td></tr> <tr> <td>SD1002</td><td>48</td><td>47</td><td>～</td><td>34</td><td>33</td></tr> <tr> <td>SD1003</td><td>64</td><td>63</td><td>～</td><td>50</td><td>49</td></tr> </table> 表中 1～64 表示站号		b15	b14	—	b1	b0	SD1000	16	15	～	2	1	SD1001	32	31	～	18	17	SD1002	48	47	～	34	33	SD1003	64	63	～	50	49	S(初始化)	
	b15	b14	—	b1	b0																														
SD1000	16	15	～	2	1																														
SD1001	32	31	～	18	17																														
SD1002	48	47	～	34	33																														
SD1003	64	63	～	50	49																														
SD1204 ～ SD1207	安全站刷新通信状态 (第 2 个 CC-Link Safety 主站模块)	0: 正常, 预约站指定, 未使用, 标准远程站 1: 安全站通信异常	- 存储安全远程站的刷新通信状态。 - 标准远程站时存储“0”。 <table border="1"> <tr> <td></td><td>b15</td><td>b14</td><td>—</td><td>b1</td><td>b0</td></tr> <tr> <td>SD1204</td><td>16</td><td>15</td><td>～</td><td>2</td><td>1</td></tr> <tr> <td>SD1205</td><td>32</td><td>31</td><td>～</td><td>18</td><td>17</td></tr> <tr> <td>SD1206</td><td>48</td><td>47</td><td>～</td><td>34</td><td>33</td></tr> <tr> <td>SD1207</td><td>64</td><td>63</td><td>～</td><td>50</td><td>49</td></tr> </table> 表中 1～64 表示站号		b15	b14	—	b1	b0	SD1204	16	15	～	2	1	SD1205	32	31	～	18	17	SD1206	48	47	～	34	33	SD1207	64	63	～	50	49	S(状态变化)	
	b15	b14	—	b1	b0																														
SD1204	16	15	～	2	1																														
SD1205	32	31	～	18	17																														
SD1206	48	47	～	34	33																														
SD1207	64	63	～	50	49																														
SD1208 ～ SD1271	安全站通信状态 (第 2 个 CC-Link Safety 主站模块)	存储与各安全远程站的通信状态	- 存储与各安全远程站的通信状态。 - SD1208: 站号 1～SD1271: 站号 64 (标准远程站、预约站指定、未连接时固定为 0) 0: 正在进行正常通信 10: 正在进行初始化 20: 正在存取内部信息 30: 链接异常 8300: 链接异常 (安全远程站检测出出错) 8310: 链接异常 (产品信息不一致) 8320: 链接异常 (初始化监视超时) 8321: 链接异常 (安全监视超时) 8322: 链接异常 (出错监视超时) 8330: 链接异常 (指令异常) 8331: 链接异常 (数据分割号异常) 8332: 链接异常 (链接 ID 异常) 8333: 链接异常 (运行号异常) 8334: 链接异常 (接收数据异常)	S(状态变化)	Q ^S																														
SD1272 ～ SD1275	安全站互锁状态 (第 2 个 CC-Link Safety 主站模块)	0: 未互锁 1: 互锁中	如果在互锁状态下主站中检测出出错, 相应站号对应的位将变为 1。 <table border="1"> <tr> <td></td><td>b15</td><td>b14</td><td>—</td><td>b1</td><td>b0</td></tr> <tr> <td>SD1272</td><td>16</td><td>15</td><td>～</td><td>2</td><td>1</td></tr> <tr> <td>SD1273</td><td>32</td><td>31</td><td>～</td><td>18</td><td>17</td></tr> <tr> <td>SD1274</td><td>48</td><td>47</td><td>～</td><td>34</td><td>33</td></tr> <tr> <td>SD1275</td><td>64</td><td>63</td><td>～</td><td>50</td><td>49</td></tr> </table> 表中 1～64 表示站号		b15	b14	—	b1	b0	SD1272	16	15	～	2	1	SD1273	32	31	～	18	17	SD1274	48	47	～	34	33	SD1275	64	63	～	50	49	S(状态变化)	
	b15	b14	—	b1	b0																														
SD1272	16	15	～	2	1																														
SD1273	32	31	～	18	17																														
SD1274	48	47	～	34	33																														
SD1275	64	63	～	50	49																														
SD1276 ～ SD1279	安全站互锁解除请求 (第 2 个 CC-Link Safety 主站模块)	0: 不解除安全站的 I/O 互锁 1: 解除安全站的 I/O 互锁	通过将本寄存器的位由 0 变为 1, 解除安全站的 I/O 互锁。 <table border="1"> <tr> <td></td><td>b15</td><td>b14</td><td>—</td><td>b1</td><td>b0</td></tr> <tr> <td>SD1276</td><td>16</td><td>15</td><td>～</td><td>2</td><td>1</td></tr> <tr> <td>SD1277</td><td>32</td><td>31</td><td>～</td><td>18</td><td>17</td></tr> <tr> <td>SD1278</td><td>48</td><td>47</td><td>～</td><td>34</td><td>33</td></tr> <tr> <td>SD1279</td><td>64</td><td>63</td><td>～</td><td>50</td><td>49</td></tr> </table> 表中 1～64 表示站号		b15	b14	—	b1	b0	SD1276	16	15	～	2	1	SD1277	32	31	～	18	17	SD1278	48	47	～	34	33	SD1279	64	63	～	50	49	U(请求时)	
	b15	b14	—	b1	b0																														
SD1276	16	15	～	2	1																														
SD1277	32	31	～	18	17																														
SD1278	48	47	～	34	33																														
SD1279	64	63	～	50	49																														

附录 3 参数号一览

在参数设置中发生了出错时参数号将被存储到特殊寄存器 (SD16 ~ 26) 中。
本节介绍参数号与参数设置位置的对应表。

附表 18 参数号一览

项目		参数号	参阅章节
标识		0000 _H	8.1 节 (1)
注释		0001 _H	
I/O 分配	类型	0400 _H	4.3 节 8.1 节 (6)
	型号		
	点数		
	起始 XY(起始 I/O 地址号)		
基本设置	基板型号	0401 _H	8.1 节 (6)
	电源模块型号		
	扩展电缆型号		
	插槽数		
组号		0.5 _{mmH}	8.2 节 (1)
开关设置		0407 _H	8.1 节 (6)
定时器时限设置	低速	1000 _H	8.1 节 (2)
	高速		9.2.8 节
RUN-PAUSE 触点		1001 _H	6.12.1 节 8.1 节 (2)
远程复位		1002 _H	6.12.2 节 8.1 节 (2)
STOP → RUN 时的输出模式		1003 _H	6.10 节 8.1 节 (2)
空插槽点数		1007 _H	4.2.1 节 8.1 节 (2)
软元件点数		2000 _H	8.1 节 (4) 9.1 节 9.2 节
WDT(看门狗定时器) 设置		3000 _H	3.2 节 8.1 节 (3)

(转下页)

附表 18 参数号一览

项目		参数号	参阅章节
恒定扫描		3003 _H	6.9 节 8.1 节 (3)
MELSECNET/H 设置	MELSECNET 个数设置	5000 _H	8.2 节 (1)
	起始 I/O 地址号	5NM0 _H	
	网络号		
	模式	5NM0 _H	
	刷新参数	5NM1 _H	
测试模式下的连续运行		6000 _H	6.5 节
远程站异常时的动作设置			8.1 节 (8)
引导文件设置		7000 _H	5.1.4 节 8.1 节 (5)
CC-Link 设置	模块个数设置	C000 _H	8.2 节 (2)
	远程输入 (RX) 刷新软元件	CNM1 _H	
	远程输出 (RY) 刷新软元件		
	远程寄存器 (RW _r) 刷新软元件		
	远程寄存器 (RW _w) 刷新软元件		
	特殊继电器 (SB) 刷新软元件		
	特殊寄存器 (SW) 刷新软元件		
	起始 I/O 地址号	CNM2 _H	
	动作设置		
	模式设置		
	传送设置		
	安全刷新监视时间		
	链接 ID		
	总链接个数		
	重试次数		
	自动恢复个数		
	扫描模式指定		
	站信息设置		

附录 4 在安全 CPU 模块中使用 MELSECNET/H 模块时的限制

- (1) 安全 CPU 模块中可设置的网络参数
- 在安全 CPU 模块中使用 MELSECNET/H 模块时，通过 GX Developer 可设置的 MELSECNET/H 的网络参数如附表 19 所示。

附表 19 通过 GX Developer 可设置的网络参数一览表

设置项目		可否设置
网络类型	MNET/H 模式 (管理站)、MNET/H 扩展模式 (管理站)	×
	MNET/H 模式 (常规站)、MNET/H 扩展模式 (常规站)	○
	MNET/10 模式 (管理站)	×
	MNET/10 模式 (常规站)	○
	MNET/H 待机站	×
	MNET/H (远程主站)	×
起始 I/O 地址号		○
网络号		○
总 (从) 站数		×
组号		○
模式		○
网络范围分配 (共用参数)		×
站固有参数		×
刷新参数		○
中断设置		×
管理站恢复设置		×
待机站对应模块		×
冗余设置		×
链接传送		×
路由参数		×
其它站存取时的有效模块		×

○：可以设置；×：不可设置

备注

关于 MELSECNET/H 的网络参数，请参阅以下手册。

➡ Q 系列 MELSECNET/H 网络系统参考手册 (PLC 网络篇)

(2) 安全 CPU 模块中可使用的 MELSECNET/H 的网络系统的功能

MELSECNET/H 网络系统的功能及安全 CPU 模块中可使用的功能如附表 20 所示。

附表 20 MELSECNET/H 网络系统的功能及在安全 CPU 模块中可否使用一览

功能		可否使用
循环传送功能	通过 LB/LW 的通信	○
	通过 LX/LY 的通信	○
	MELSECNET/H 扩展模式	○
	刷新参数	○
	共用参数	○
	站固有参数	×
	数据链接传送功能	×
	预约站指定	○
	低速循环传送功能	○
	冗余系统功能	×
瞬时传送功能	通信功能	×
	路由功能	×
	组功能	×
	使用了逻辑通道号的信息发送功能	×
	数据发送 / 接收 (SEND/RECV)	×
	其它站字软元件读取 / 写入 (READ/SREAD/WRITE/SWRITE)	×
	其它站瞬时请求 (REQ)	×
	其它站字软元件读取 / 写入 (ZNRD/ZNWR)	×
	远程 RUN/ 远程 STOP (RRUN/RSTOP)	×
	其它站 CPU 模块的时钟数据的读取 / 写入 (RTMRD/RTMWR)	×
RAS 功能	自动恢复功能	○
	管理站转移功能	×
	管理站恢复控制功能	×
	回送功能 (光回路系统)	○
	外部供应电源引起的宕站发生的防止 (光回路系统)	○
	站脱离功能 (同轴总线系统)	○
	即使 CPU 异常时也可瞬时传送	×
	瞬时传送的异常检测时间的确认	×
	诊断功能	○
链接软元件的直接存取		×
中断顺控程序的启动		×
多重传送功能 (光回路系统)		○
网络的简易冗余		×
循环传送的停止 / 再启动以及链接刷新停止 (网络测试)		○
由于同一网络号模块的多个安装引起的发送点数的增加		×
多 CPU 系统对应		×
远程 I/O 系统		×
冗余系统对应		×
网络诊断 (线路监视)		○

○ : 可以使用 ; × : 不可使用

备注

.....

关于 MELSECNET/H 网络系统的功能，请参阅以下手册。

 Q 系列 MELSECNET/H 网络系统参考手册 (PLC 网络篇)

.....

附录 5 电池运输时的注意事项

运输含锂的电池时，需要按运输规定进行处理。

(1) 规定对象种类

QS 系列 CPU 模块中使用的电池的分类如附表 21 所示。

附表 21 运输规定对象种类一览表

产品名称	型号	产品形态	运输处理
电池	Q6BAT	锂电池单体	非危险物

(2) 运输时的处理

产品出厂时三菱公司已按运输规定进行了包装，当客户重新包装或打开包装后进行运输时，应按照 IATA Dangerous Goods Regulations(IATA 危险品规范)、IMDG Code(国际海上危险品运输规程)以及各国的运输规定进行运输。

此外，详细情况应与运输商确认。

索引

[0] ~ [9]

- 10 进制数 9-34
- 16 进制数 9-34

[A]

- 安全 9-26
- 安全设置 8-9

[B]

- B(链接继电器) 9-14
- BCD(2 进制编码的 10 进制数) 3-19
- BIN(2 进制数) 3-16
- 报警器 (F)
 - OFF 时的处理 9-11
 - ON 时的处理 9-10
- 报警器 [F] 9-9
- 变址继电器 (V) 9-13
- 步骤
 - 引导运行的~ 5-9

[C]

- C(计数器) 9-23
- CC-Link 设置 8-12
- 参数
 - 可编程控制器~ 8-2
 - 网络~ 8-10
- 程序存储器 5-3
- 程序容量 2-1
- 程序语言 2-1
- 出错
 - ~解除 6-30
 - ~发生时的 LED 显示 6-30
- 存储器构成 5-1
- 存储器容量 5-16

[D]

- D(数据寄存器) 9-27
- 低速定时器 (T) 9-17
- 低速累计定时器 (ST) 9-19
- 电池运输 附录 -24
- 定时器 (T) 9-17
 - 处理方法 9-20
 - 精度 9-21
- 定时器的计测单位 9-18
- 定时器时限设置 8-3

[F]

- F(报警器) 9-9

[G]

- 高速定时器 (T) 9-18
- 高速累计定时器 (ST) 9-19
- 格式化 5-3

- 功能版本的确认方法 1-9
- 功能一览 6-1

[H]

- H(16 进制数) 9-34
- HEX(16 进制数) 3-18
- 恒定扫描 6-36

[I]

- I/O 地址号 4-1
- I/O 地址号的分配 4-2
- I/O 点数 2-1
- I/O 分配的目的 4-5
- I/O 分配的思路 4-6
- I/O 分配设置 8-7
- I/O 控制方式 2-1
- I/O 软件点数 2-1

[J]

- 计数器 (C)
 - 计数器 (C) 9-23
 - ~的复位 9-25
 - 计数处理 9-23
- 继续运行出错 6-27
- 监视 6-51
- 看门狗定时器 (WDT) 6-56

[K]

- K(10 进制数) 9-34
- 可编程控制器参数 8-2
- 可编程控制器名称设置 8-2
- 可编程控制器系统设置 8-3
- 可存储的数据 5-2
- 空插槽点数 8-3
- 扩展名 5-2

[L]

- LED
 - ~的熄灯方法 6-60
 - ~的显示 6-60
- 累计定时器 9-19
- 链接继电器 (B) 9-14
- 链接寄存器 (W) 9-28
- 链接特殊继电器 (SB) 9-16
- 链接特殊寄存器 (SW) 9-30

[M]

- M(内部继电器) 9-8
- MELSECNET/H 设置 8-11

[N]

- N(嵌套) 9-33
- 内部继电器 (M) 9-8

内部系统软元件	9-31
内部用户软元件	9-2
内置 ROM	5-6

[P]

PC RAS 设置	8-4
-----------------	-----

[Q]

嵌套 (N)	9-33
嵌套参数	8-10
清除	
计数值的~	9-25
累计定时器的~	9-19
驱动器号	5-2

[R]

RUN/PAUSE 触点	2-2
RUN 状态	3-9
容量 (文件容量)	5-16
软元件点数	
报警器 (F)	2-2
变址继电器 (V)	2-2
定时器 (T)	2-2
计数器 (C)	2-2
累计定时器 (ST)	2-2
链接继电器 (B)	2-2
链接寄存器 (W)	2-2
链接特殊继电器 (SB)	2-2
链接特殊寄存器 (SW)	2-2
内部继电器 (M)	2-2
数据寄存器 (D)	2-2
特殊继电器 (SD)	2-2
特殊寄存器 (SM)	2-2
软元件设置	8-5
软元件一览	9-1

[S]

SB(链接特殊继电器)	9-16
SD(特殊寄存器)	9-23
SM(特殊继电器)	9-31
ST(累计定时器)	9-19
STOP → RUN 时的输出模式	6-39
STOP 状态	3-9
SW(链接特殊寄存器)	9-30
扫描时间	3-5
扫描时间的精度	3-5
时钟功能	6-24
时钟数据的读取	6-43
时钟数据的更改	6-43
是否需要格式化	5-2
输入 (X)	9-5
数据的清除处理	3-12
数据寄存器 (D)	9-27
顺控程序	3-2
瞬时掉电	3-11

[T]

T(定时器)	9-17
特点	1-2
特殊继电器 (SM)	9-31
特殊寄存器 (SD)	9-32
停止出错	6-27

[V]

V(变址继电器)	9-13
------------------	------

[W]

W(链接寄存器)	9-28
WDT(看门狗定时器)	6-56
WDT 设置	6-56
文件	5-13
文件的存储器容量	5-16
文件的大小单位	5-17

[X]

X(输入)	9-5
X/Y 分配确认	8-8
系统监视	6-58
写入	
至程序存储器的~	5-5
序列号	1-9
序列号的确认方法	1-9

[Y]

Y(输出)	9-7
引导文件设置	8-6
引导运行	5-8
用户设置的系统区	5-4
远程 RESET(远程复位)	6-48
远程 RUN/STOP	6-45
运行中写入	6-52
运行中写入预留容量 (步)	5-13

[Z]

注意事项	
安装时的~	A-2
废弃时的~	A-5
连线时的~	A-3
启动、维护时的~	A-4
设计方面的~	A-1
运输时的~	A-6
自诊断	6-27
最大计数速度	9-26

质保

三菱安全可编程控制器质保条款

1. 质保及产品支持

- (1) 质保期限：三菱电机公司（简称三菱）的三菱安全可编程控制器（本产品）的免费质保期限为自购买日起或货到指定地点日起的 1 年内、或者从产品制造日起 18 个月内中的最先到达的期限。
- (2) 质保内容：三菱认定为本产品的故障时，将从以下的 4 个方式中选择一个三菱认为最合适的方式实施质保：本产品的无偿维修、无偿更换、购买金额的折扣或者购买价格的全额退款。
- (3) 质保生效的必要手续：用户如果未按以下各条目履行质保的申请手续，三菱将不对上述 1. (2) 中记载的本产品的质保责任负责。以下手续为使本产品的质保生效的前提条件，因此务必加以注意。
 - 1) 质保上的索赔的书面通知：在通知了本产品的质保后 30 日内，应向三菱以及购得本产品的代理店或者销售商递交用户产品质保方面问题的详细内容。此外，对于超过了上述 1. (1) 中规定的质保期限的通知，除以下 1. (5) 中相应的有偿维修以外，将不予受理。必须在质保期限内按照规定进行通知。
 - 2) 针对用户索赔申请的本产品检查方面的用户协助义务：三菱对用户质保索赔进行调查时用户应予以协助。协助的内容包括：对应于索赔内容的本产品的状态及原因证据的保存、针对三菱询问的回答、用户持有的记录的提供，在三菱认为需要进行本产品的工厂试验或者安装位置下的试验时，相应试验的允许等。
 - 3) 运费的承担：在进行用户的质保索赔的原因调查时，或者发现本产品故障情况下的维修或更换时，有时三菱会委托用户拆卸相应产品并寄送至三菱或者三菱代理商所在地。此时发生的拆卸费用、往返运输费及维修、更换、本产品的再安装等费用应由用户承担。
 - 4) 出差维修费用的承担：无论是到国内还是国外，三菱接受用户请求派遣出差维修人员以及部件运输所耗费用应由用户承担。但是，对于包括本产品的维修、更换在内的再安装、现场调试、维护保养或者现场试验，三菱不负责任。
- (4) 日本国外的维修：在海外是由三菱指定的各地地区的 FA 中心受理维修事宜。但是，对于三菱的质保范围以外的维修服务，根据各 FA 中心的情况其维修费用及维修条件等将有可能不同。
- (5) 有偿维修：即使是在上述质保期结束后，三菱将在产品停产后的 7 年内受理本产品的有偿维修，但仅限于三菱有库存备件的情况下。当产品停产时，三菱通常会生产和保留足够的备用部件，以便提供 7 年的产品维修服务。此外，受理有偿维修时的合同条件是受理有偿维修申请时有效的三菱的标准有偿维修条件。
- (6) 关于产品停产：产品停产的消息将以三菱技术公告等方式予以通告。对于产品停产后的本产品供应（包括备件），有可能发生无法供应的情况。

2. 质保范围

- (1) 对于包括安全系统、失效保障系统、紧急停止系统在内的使用本产品的设备、系统或者生产线的材质、建筑基准、功能、使用、特性、其它性质的任何保证、设计、制造、建筑、安装等，三菱均不负责。
- (2) 对于使用本产品的应用、设备或者系统中合适的安全系数及冗余度的确定，本产品是否适用于用户想要实现的特定目的、用途的确定，三菱将不负责任。
- (3) 用户使用本产品时，对于本产品的适用性、应用、设计、结构以及安装及调整的正确与否的判断，应由具有三菱指定的培训课程结业资格的或者具有与此相当的经验的专业技术人员进行。
- (4) 在将本产品安装在用户或最终用户的设备、生产线或系统中组合使用时，关于产品的功能适用性以及是否符合应用标准和要求，三菱公司不负责设计和进行测试。
- (5) 以下情况下，即使在免费质保期内，也不能作为质保对象。
 - 1) 由除三菱或三菱授权的 FA 中心以外的人员进行过维修或改造。
 - 2) 由于用户过失、疏忽、事故、不当使用而受到过损伤。
 - 3) 由于用户不当的存储、操作、安装或维护而造成的故障。
 - 4) 由于不正确的设计、与不兼容或存在缺陷的硬件或软件组合使用而造成的故障。
 - 5) 如果正确维护或更换了使用手册中指定的耗材（电池、背光灯、保险丝等）后本可以避免的故障。
 - 6) 由于安装了本产品的设备、生产线或系统不符合相应的法律、安全和行业标准而造成的产品故障。
 - 7) 将本产品用于异常的应用中。
 - 8) 在安装、操作或使用本产品时违反了三菱的产品用户手册、说明书、安全手册、技术公告和指南中所介绍的用法说明、注意事项或警告而造成的故障。
 - 9) 根据本产品出厂时的科技水准无法预知的故障。
 - 10) 由于使用在过热、潮湿、异常电压、冲击、过度振动、物理损坏等不适当的环境中而造成的故障。
 - 11) 由于地震、暴风、水灾等不可抗力、火灾、故意破坏、犯罪、恐怖事件、通讯或电源故障等其它三菱无法控制的状况所造成的故障。
- (6) 三菱主页上和三菱提供的产品目录、手册或技术资料中记载的产品信息和规格如有改变，恕不另行通知。
- (7) 三菱主页上和三菱提供的产品目录、手册、技术公告或其它资料中记载的产品信息和说明仅作为用户使用本产品时的指南，并不作为产品销售时的保证，也不作为产品销售合同的一部分。
- (8) 本质保条款上的各条件包含了用户与三菱之间关于质保、补偿措施及损害赔偿的所有意向，应优先于两当事者之间的无论书面或口头上的任何其它事前意向。
- (9) 三菱仅提供本条款中记载的有关本产品的质保和补偿措施，对除此以外的任何其它质保和补偿措施不予提供。

3. 质保的上限

- (1) 对于用户提出的质保违约、合同违约、过失、严重民事侵权以及本产品的销售、维修、退换、配送、性能、状态、适用性、可靠性、安装、使用等方面的索赔，三菱的关于本产品的最大累计法律责任赔偿额以本产品的价格为上限。
- (2) 尽管三菱已经取得了德国 TUV Rheinland 的国际安全标准 IEC61508 和 EN954-1/ISO13849-1 的产品可靠性认证，但这并不保证本产品不发生任何故障。本产品的用户应遵守所有现行的安全标准、规则或法律，并应对本产品所安装或使用的系统采取

适当的安全措施，除了本产品之外还应当同时采取其它的安全措施。对于如果遵守了现行的安全标准、规则或法律则可以预防的损害，三菱不负任何责任。

- (3) 三菱禁止将本产品用于电厂、火车、铁路系统、飞机、航空管理、其它运输系统、娱乐设备、医院、医疗、透析和生命维持设备、焚化和燃烧设备、原子能、危险品或化学品处理、采矿和冶炼等可能涉及人员生命健康安全和重大财产安全的系统。
- (4) 对于特殊损失、利润 / 销售 / 收入损失、工作量和成本的增加、生产停工的损失、成本超限、环境污染损害赔偿及包含清污成本在内的附带的或间接的损失，无论损失是否基于合同违约、质保违约、法令违反、过失或其它民事责任，三菱均不承担责任。
- (5) 在针对三菱提出的由于产品或其缺陷所导致的损害事件中，对于造成人身伤害、意外死亡或物质性财产损失这三类损失的全部范围，本质保条款中的拒绝和限制将服从法律的规定。因此，对于这类法律规定的损失，即使条款中存在拒绝和限制性规定，也可遵照法律对这类损失进行强制执行。
- (6) 对于质保违约或其它关于本产品的问题，购买本产品的用户应当自购买之日起一年内提出。
- (7) 本质保条款中记载的三菱的责任限制，对用户的索赔的补偿方法、损害赔偿等的条件全部是个别独立具有强制力的意向事项，任何包含构成用户与三菱之间的买卖合同的质保条件、约束、损害赔偿的上限的意向事项都不具有法律的强制力，以后即使由法庭作出了判决，对剩余的条款的有效性或者强制执行可能性也不产生影响。

4. 交货 / 不可抗力

- (1) 三菱承认的产品交货日期为估算日期，而非承诺的交货日期。三菱将尽一切努力根据用户订单上或购买合同上规定的交货日程按时交货，但如不能按时交货将不承担损害赔偿的责任。
- (2) 由于某种事由用户希望延迟收货时，所发生的相应保管费用、拒绝或延迟收货产生的风险及费用应由用户承担。
- (3) 对于因原材料的不足、零件供应商的交货延迟、所有劳动纠纷、地震、火灾、暴风、水灾、偷盗、犯罪、恐怖活动、战争、禁运、政府规定、运输中途损失或耽搁、不可抗力等原因，或者三菱无法控制的其它情况所造成的产品损失、交货 / 服务 / 维修 / 退换延迟等，三菱将不承担责任。

5. 法律的选择

如果对本质保条款以及用户与三菱之间的任何协定或合同发生争议，应选择产品安装所在地的相关法律作为裁判依据。

6. 仲裁

与本产品及其销售和使用有关的任何争议或主张，可通过产品安装所在地的仲裁机构进行仲裁。

SH (NA) -080713CHN-A (0708) MEACH

MODEL: QSCPU-U-KP-C



地址：上海市虹桥路1386号三菱电机自动化中心

邮编：200336

电话：021-23223030 传真：021-23223000

网址：<http://cn.MitsubishiElectric.com/fa/zh/>

技术支持热线 **400-821-3030**



扫描二维码,关注官方微博



扫描二维码,关注官方微信

内容如有更改 恕不另行通知